DEEP LEARNING FOR PREDICTION OF DRUG-TARGET INTERACTION SPACE AND PROTEIN FUNCTIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET SÜREYYA RİFAİOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JUNE 2020

Approval of the thesis:

DEEP LEARNING FOR PREDICTION OF DRUG-TARGET INTERACTION SPACE AND PROTEIN FUNCTIONS

submitted by **AHMET SÜREYYA RİFAİOĞLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of the Department, **Computer Engineering** _____

Prof. Dr. Mehmet Volkan Atalay
Supervisor, **Computer Engineering, METU** _____

Prof. Dr. Rengül Çetin-Atalay
Co-Supervisor, **Department of Health Informatics, METU** _____

Examining Committee Members:

Prof. Dr. Tolga Can
Computer Engineering, METU _____

Prof. Dr. Mehmet Volkan Atalay
Computer Engineering, METU _____

Prof. Dr. Erden Banoğlu
Faculty of Pharmacy, Gazi Uni. _____

Assoc. Prof. Dr. Yakup Kutlu
Computer Engineering., İskenderun Technical Uni. _____

Assoc. Prof. Dr. Tunca Doğan
Computer Engineering, Hacettepe Uni. _____

Date: 30.06.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ahmet Süreyya Rifaioğlu

Signature :

ABSTRACT

DEEP LEARNING FOR PREDICTION OF DRUG-TARGET INTERACTION SPACE AND PROTEIN FUNCTIONS

Rifaioğlu, Ahmet Süreyya
Doctor of Philosophy, Computer Engineering
Supervisor: Prof. Dr. Mehmet Volkan Atalay
Co-Supervisor: Prof. Dr. Rengül Çetin-Atalay

June 2020, 224 pages

With the advancement of sequencing and high-throughput screening technologies, large amount of sequence and compound data have been accumulated in biological and chemical databases. However, only small number of proteins and compounds have been annotated by wet-lab experiments due to the huge compound and chemical space. Therefore, computational methods have been developed to annotate protein and compound space. In this thesis, we describe the design and implementation of several methods for accurate drug-target interaction prediction and functional annotations of proteins within the framework of Comprehensive Resource of Biomedical Relations with Deep Learning and Network Representations (CROssBAR) project whose aim is to integrate biological and chemical data scattered in different sources and to create prediction methods for drug discovery based on deep learning. The first method, DEEPred is a sequence based automated protein function prediction method that employs a stacked multi-task deep neural networks based on Gene Ontology (GO) directed acyclic graph hierarchy. The performance of DEEPred was compared with state-of-the-art methods and its source code is available at https://github.com/cansyl/deepred. DEEPScreen is the second method and it is a drug-target interaction (binary) prediction method. In

DEEPScreen, the idea is to learn compound features automatically using compound images via convolutional neural networks. DEEPScreen was trained for 704 target proteins and the input compounds predicted as active or inactive against trained targets. The performance of DEEPScreen was compared with the state-of-the art methods using different benchmarking datasets. The source code is available at https://github.com/cansyl/DEEPScreen. The third method is called MDeePred which is a binding affinity prediction method. MDeePred is a chemogenomic method where both protein and compounds features were fed to a hybrid pairwise deep neural network structure. The main difference between MDeePred and DEEPScreen in terms of features is that MDeePred employs compound-target feature pairs whereas in DEEPScreen only compound features were used. The main novelty of MDeePred is the proposed multi-channel featurization approach for protein sequences where each channel represents a different property of input protein sequences. The performance of MDeePred was calculated on multiple benchmarking datasets and compared its performance with the state-of-the-art methods. The source code for MDeePred is available at https://github.com/cansyl/MDeePred. The fourth method is called iBioProVis which is an online interactive visualization tool for chemical space. The main purpose of iBioProVis is to embed and visualize compound features on 2-D space. It relies on the assumption that topologically and chemically similar compounds have similar bioactivity profiles. The inputs for iBioProVis are target protein identifiers and optionally, SMILES strings of user-input compounds. The tool then generates circular fingerprints for active compounds of targets and user-input compounds and then, t-Stochastic Neighbor Embedding (t-SNE) method is used to embed compounds on 2-D space. The tool also provides cross-references for well-known databases for input targets and compounds. iBioProVis is available at https://ibioprovis.kansil.org/.

Keywords: Virtual Screening, Deep Learning, Protein Function Prediction, Binding Affinity Prediction, Drug-Target Interaction Prediction

ÖZ

İLAÇ-HEDEF PROTEİN ETKİLEŞİM UZAYI VE PROTEİN FONKSİYONLARININ TAHMİNİ İÇİN DERİN ÖĞRENME

Rifaioğlu, Ahmet Süreyya
Doktora, Bilgisayar Mühendisliği
Tez Yöneticisi: Prof. Dr. Mehmet Volkan Atalay
Ortak Tez Yöneticisi: Prof. Dr. Rengül Çetin-Atalay

Haziran 2020, 224 sayfa

Sekanslama ve yüksek çıktılı tarama teknolojilerinin ilerlemesi ile biyolojik ve kimyasal veri tabanlarında büyük miktarda protein ve bileşik verisi birikmiştir. Bununla birlikte, protein ve bileşik uzaylarının büyüklüğü sebebiyle bu verilerin çok azı laboratuvar deneyleriyle anlamlandırılabilmiştir. Bu nedenle, protein ve bileşik uzayını anlamlandırılabilmek için hesaplamalı yöntemler geliştirilmektedir. Bu tezde; amacı farklı kaynaklardaki biyolojik ve kimyasal verileri birleştirmek ve ilaç keşfi için derin öğrenme tabanlı yöntemler geliştirmek olan Biyomedikal İlişkilerin Kapsamlı Kaynağı ve Biyomedikal İlişkileri (CROssBAR) projesi kapsamında, ilaç-hedef protein etkileşimi tahmini ve proteinlerin fonksiyonel anlamlandırılması için çeşitli yöntemlerin tasarlanması ve uygulanmasını tarif ediyoruz. İlk yöntem olan DEEPred, Gen Ontoloji'sinin yönlü düz ağaç hiyerarşisine dayanan ve yığılmış çok görevli derin sinir ağlarını kullanan protein fonksiyon tahmin yöntemidir. DEEPred'in performansı, literatürdeki iyi bilinen yöntemlerle karşılaştırılmıştır ve kaynak kodu https://github.com/cansyl/deepred adresinde bulunmaktadır. Geliştirilen ikinci yöntem, ilaç-protein hedefi etkileşimi (ikili) tahmin yöntemi olan DEEPScreen'dir. DEEPScreen'deki ana fikir, evrişimli sinir ağları aracılığıyla

bileşik görüntülerini kullanarak özelliklerini otomatik olarak öğrenmektir. DEEPScreen, 704 hedef protein için eğitilmiş ve girdi bileşikleri, eğitilmiş hedeflere karşı aktif ya da inaktif olarak tahmin edilmiştir. DEEPScreen'in performansı, farklı kıyaslama veri kümeleri kullanılarak literatürdeki yöntemlerle karşılaştırılmıştır. Yöntemin kaynak kodu https://github.com/cansyl/DEEPScreen adresinde bulunmaktadır. Üçüncü yöntem olan MDeePred protein-bileşik bağlanma değeri tahmini yöntemidir. MDeePred, hem protein hem de bileşik özelliklerinin çift girdili melez derin sinir ağı yapısına beslendiği kemogenomik bir yöntemdir. Girdi olarak kullanılan özellikler açısından MDeePred ve DEEPScreen arasındaki temel fark, MDeePred'in bileşik hedef özellik çiftlerini kullanması, bunun yanında DEEPScreen'de sadece bileşik özelliklerinin kullanılmasıdır. MDeePred'in sunduğu ana yenilik, her bir kanalın girdi protein dizilerinin farklı bir özelliğini temsil ettiği çok kanallı özelliklendirme yaklaşımıdır. MDeePred'in performansı birden fazla kıyaslama veri kümesinde hesaplanmış ve performansı literatürde iyi bilinen yöntemlerle karşılaştırılmıştır. MDeePred'in kaynak kodu https://github.com/cansyl/MDeePred adresinde yer almaktadır. Dördüncü yöntem olan iBioProVis, kimyasal uzay için çevrimiçi ve etkileşimli bir görüntüleme aracıdır. iBioProVis'in temel amacı, bileşik özellikleri 2 boyutlu uzaya yerleştirmek ve bu bağlamda bileşikleri görselleştirmektir. Bu araç, topolojik ve kimyasal olarak benzer bileşiklerin benzer biyoaktivite profillerine sahip olduğu varsayımına dayanır. iBioProVis için girdiler, hedef protein tanımlayıcıları ve isteğe bağlı olarak kullanıcılar tarafından verilen bileşiklerinin SMILES gösterimleridir. Araç, daha sonra hedeflerin aktif bileşikleri ve kullanıcı girdi bileşikleri için dairesel parmak izlerini üretir ve bileşikleri 2 boyutta göstermek için t-Stokastik Yakınlık Gömmesi yöntemi kullanılır. Aynı zamanda, girdi hedef proteinleri ve bileşikleri için iyi bilinen veri tabanları için çapraz referanslar sağlanmaktadır. iBioProVis'e https://ibioprovis.kansil.org/ adresinden ulaşılabilir.

Anahtar Kelimeler: Sanal Tarama, Derin Öğrenme, Protein Fonksiyonu Tahmini, Bağlanma Değeri Tahmini, İlaç-Hedef Protein Etkileşimi Tahmini

To İnci, Semih and Çınar

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my supervisor, Dr. Volkan Atalay and Dr. Rengül Çetin-Atalay. They always guided and encouraged me to be a good scientist. Without their positive and persistent help and support, this thesis would not have been realized.

I am deeply indebted to Dr. Tunca Doğan for being a great collaborator and for all the work we carried out together as a team in CROssBAR project. I also thank my friends Murat and Abdullah who were with me from beginning of my academic carrier. I am grateful to my teammates/roommates Gökhan, Alperen and Ahmet for being not only my colleagues but also being great friends.

This thesis supported by Comprehensive Resource of Biomedical Relations with Deep Learning and Network Representations (CROssBAR) project which is funded through the Newton/Katip Celebi Institutional Links program by TÜBİTAK, Turkey and British Council, UK (project no: 116E930). I am also thankful to TÜBİTAK for accepting my research proposal and supporting my visit to European Bioinformatics Institute for 6 months through 2214 International Doctoral Research Fellowship Programme. I am grateful to Dr. Maria Jesus Martin for inviting me to EMBL - European Bioinformatics Institute and gave me the opportunity to work with UniProt Automatic Annotation team.

I wish to express my gratitude and thank my wife İnci for her love and constant support all these years. Thank you for being my best companion in this life. My sons Semih and Çınar, you have no idea how you changed me positively as a human-being. You made me laugh and supported me emotionally with your pure love when I was having tough and stressful times. I cannot express my appreciation to my father, my mother and my sisters for helping and supporting me in every part of my life. I love you all.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional neural networks -
DTI	Drug-Target Interaction prediction
GO	Gene Ontology
VS	Virtual Screening
PCM	Proteochemometric modeling
SMILES	Simplified molecular-input line-entry system
InChI	International Chemical Identifier
CAFA	Critical Assessment of Functional Annotation
PINN	Pairwise Input Neural Network
EC	Enzyme Commission
ECFP	Extended Connectivity Fingerprints
FDA	Food and Drug Administration
DNN	Deep Neural Networks
GPCR	G-Protein Coupled Receptor
RNN	Recurrent Neural Network
DBN	Deep Belief Network
RBM	Restricted Boltzmann Machines
MUV	Maximum Unbiased Validation
DUD-E	A Database of Useful (Docking) Decoys - Enhanced
ANN	Artificial neural networks
GCN	Graph convolutional network

CHAPTER 1

INTRODUCTION

[1]The development of new drugs and understanding functions of proteins are the key problems and challenges to improve the current field of biomedicine. Computational methods have been used in bioinformatics and cheminformatics studies for nearly three decades, to aid discovering the molecular mechanisms of proteins and to propose novel treatment options for several diseases. Recent advances in computational power (e.g., massively parallel and GPU computing) and in data analysis and inference techniques (e.g., artificial intelligence, machine learning, deep learning) provide opportunities for various fields including biomedicine.

My thesis is done as a part of Comprehensive Resource of Biomedical Relations with Deep Learning and Network Representations (CROssBAR) project which is funded by TÜBİTAK and British Council. The first aim of CROssBAR project is to integrate biological and chemical data coming from different resources in the context of drug discovery. The second aim of the project is to create computational drug discovery and repositioning methods based on deep learning algorithms. The last objective of the project is the generation of multi-partite networks where nodes will represent different entities and edges will relations between biological and chemical entries.

The terminology used in this thesis is given below:

- A *ligand* is a molecular structure that physically binds another molecular structure and modulates its function.
- A *compound* is a chemical structure that is formed by the combination of two or more atoms that are connected by chemical bonds.

[1] Some parts of this chapter were published in *Briefings in Bioinformatics* journal in 2019 [7]. Please note that only the parts that I worked on were included from our publication.

- Some of the compounds, *bioactive compounds*, modulates the functions of bio-molecules such as proteins.
- A *drug* is an approved (by Food and Drug Administration, for example) bioactive compound that acts on protein *targets* to cure/decelerate a specific disease or to promote the health of a living being.
- A *target protein* (or just a *target*) is a naturally occurring bio-molecule of an organism that is bound by a *ligand* and have its function modulated, which results in a physiological change in the body of the organism.
- *The Anatomical Therapeutic Chemical (ATC) Classification System* is a controlled vocabulary to classify drugs hierarchically based on their therapeutic, pharmacological and chemical properties. There are five levels in each ATC code and each level of an ATC code represents a different property of drugs. The first level represents anatomical groups; the second level shows a therapeutic main group; the third level represents a therapeutic and pharmacological subgroup; the fourth level represents a chemical, therapeutic and pharmacological subgroup; and the fifth level shows the indicated chemical substance.
- *Cheminformatics* is the application of computational techniques to the field of chemistry. Most of the virtual screening methods are considered to be cheminformatics-based.
- Gene Ontology (GO) is a controlled vocabulary that represents relationship among functions and locations of proteins in the form of directed acyclic graph structure where each node (i.e. GO term) represents a different property. GO has three categories which are molecular function, biological process and cellular component. Molecular function GO terms represent events occur at molecular level within the cells. Biological process GO terms denote series of events happening in the cells. Finally, cellular component GO terms show subcellular compartments.

It is important to note that, “small molecule”, “ligand” and “compound” were used synonymously to refer to the “chemical substances”. The term “bioactive

compound" corresponds to chemical substances with biological activities. The term "ligand" represents a chemical substance that interact with a target biomolecule to accomplish a biological purpose. The term "drug" is used to represent approved bioactive compounds, which are currently being used in the clinics. "Active pharmaceutical ingredients" (APIs) refers to the biologically active ingredient in a drug, and are responsible for the interactions with cellular polymeric macromolecules as well as small secondary messenger molecules. The terms "biomolecule", "receptor", "target" and "protein" refer to the cellular biological molecules targeted by APIs and/or bioactive compounds.

In terms of the statistics, there are tens of millions of compounds available in compound and bioactivity databases [1]–[4]. There are about 9,000 FDA approved small molecule drugs (approved + experimental) [5], roughly 560,000 reviewed protein records available (20,244 of which are human proteins) in protein sequence and annotations resources (e.g., UniProtKB/Swiss-Prot) and nearly 2,700 of human proteins are known to be targeted by either approved or experimental drugs [1], [6].

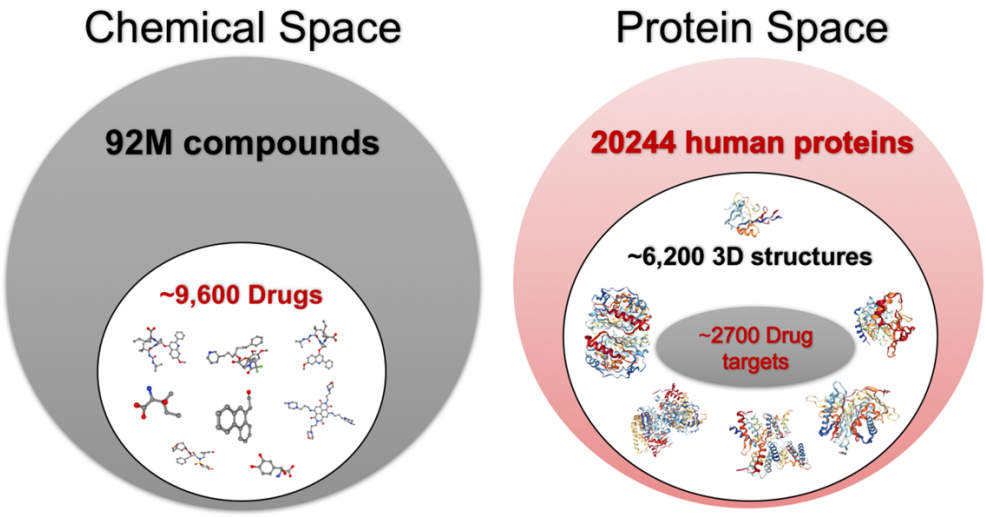


Figure 1.1. Statistics of current chemical and protein spaces in open access chemical and biological data repositories (Adapted from our publication [7]).

3D structure information of proteins and compounds provide important qualities of these molecules to determine their functions and bioactivities. However, 3D

structures of a relatively small sub-set of compounds (i.e., around 24,000) and human proteins (i.e., about 6,200) are experimentally known (partly or completely) and currently available in Protein Data Bank - PDB (Figure 1.1) [5].

The main *role* of drugs, which are bioactive compounds, is the alteration of cellular events involved in disease conditions for treatment purposes. The following two problems are of importance for the hit discovery, one of the initial steps in the development of new drugs:

- identification of novel bioactive compounds for a target protein,
- and identification of new targets for known bioactive compounds.

Drug discovery is defined as the process of identifying the roles of bioactive compounds to develop new drugs, and it is usually one of the initial steps in a drug development pipeline. Traditionally, drug research and development starts with the identification of the biomolecular targets for an intended treatment and it proceeds with the high-throughput screening experiments to identify bioactive compounds for the defined targets, together with the corresponding bioactivity levels. The aim of high-throughput screening is to find suitable drug candidates. With the advancement of high-throughput screening technology, it is now possible to conduct experiments to scan thousands of different compounds and detect their bioactivity levels on selected target proteins [8]. However, designing high-throughput screening experiments is expensive, it is a time-consuming process and it requires advanced laboratories having chemical and biological libraries. Furthermore, it is not feasible to conduct high-throughput screening experiments for all expressed proteins in the human genome and for all known compounds [9]. Another problem with high-throughput screening is its high failure rates, which limits the identification of novel drugs [10]. The problem escalates when we consider the process of drug development. The term drug development refers to the whole process to bring a drug to the market, starting with the drug discovery and ending with clinical trial phases. In Figure 1.2, main phases of the drug development procedure are shown. Most of the drug candidates fail to become an approved drug in the late phases of clinical trials due to the unexpected side effects and toxicity problems. In 2010, the cost of

developing a single drug was estimated about 1.8 billion US dollars and the process requires about 13 years on average [9]. Another important topic in drug discovery is drug repurposing whose aim is to find new usages of existing drugs. Drug repurposing significantly decreases duration of the overall drug discovery process as the repurposed drugs have already passed the main phases.

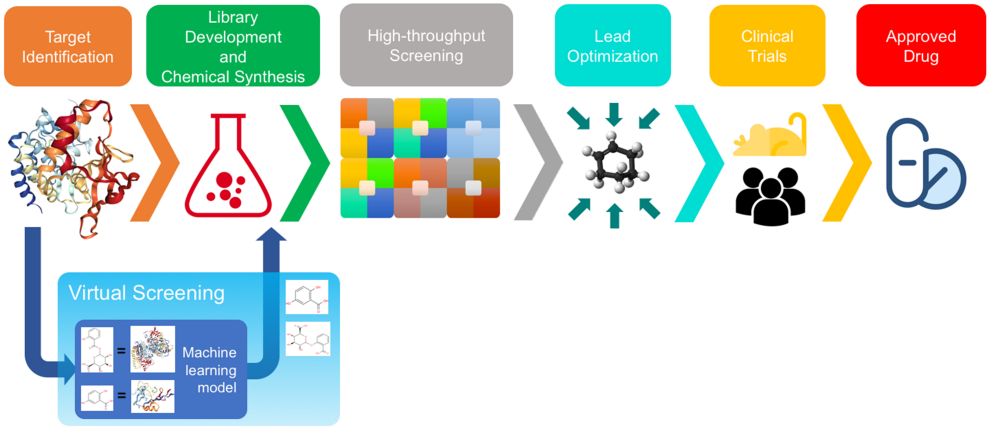


Figure 1.2. A broad overview of drug development and the place of virtual screening in this process (Adapted from our publication [7]).

Proteins play many important roles in the cells of living beings such as catalyzing cellular events, transferring other molecules and so on. Therefore, identification of functions and subcellular localizations of proteins are essential to understand inner mechanisms of cellular events within cells. For example, determining subcellular locations of targeted proteins should be known to develop drugs, so that drugs could reach the subcellular location that targeted proteins exist and bind them. Protein functions are traditionally determined by wet-lab experiments and many biocurators are working to annotate protein in terms of their functions by extracting relevant information from scientific papers. However, number of protein sequences in protein databases is rapidly growing and traditional methods cannot keep up with this growth. For example, in the current release (2020_05) of UniProt Knowledgebase (UniProtKB), there are about 181 million protein sequences and only 562,253 of them have been manually curated by biocurators [11].

1.1 Problem Statement

The ability of traditional machine learning to extract features from its raw data is quite limited. Therefore, in standard machine learning applications, the majority of the time is spent on careful preprocessing and feature engineering steps based on domain expertise with the aim of extracting meaningful and useful features to be fed to machine learning algorithms. Only then, these traditional machine learning algorithms can be used to create predictive models for certain tasks. Deep learning algorithms are capable of extracting complex features from raw input data. They create multiple levels of representations of raw input data by applying non-linear operations at each layer and they inherently learn and extract complex features automatically from raw data. Although the history of deep learning algorithms goes back to decades ago, these algorithms could not be applied due to the limitations in hardware technologies and the problems such as overfitting and lack of efficient implementations. The advancement in hardware technologies and the development of highly efficient deep learning frameworks have enabled the applications of deep learning algorithms in several fields in recent years and these algorithms have been shown to outperform the state-of-the-art methods in the fields such as computer vision, natural language processing and bioinformatics.

Several computational methods have been developed in the last decades for both computational drug discovery and protein function prediction problems. In protein function prediction, the aim is to infer functions of proteins by using different features of proteins such as their sequence properties, domains etc. The input is generally protein sequences and computational methods are used to extract features from the input sequences. The field of *in silico* estimation of unknown drug-target pairs using statistical models is called virtual screening -VS- (i.e., drug-target interaction -DTI- prediction). In drug development pipelines, VS methods are mostly placed just before the high-throughput screening, so that the unlikely drug-target pairs are eliminated; as a result, only potentially active combinations are run through the experimental screening procedure (Figure 1.2). The results published so far

shows that there are still room for significant improvements in these areas. In this thesis, I proposed solutions to four different problems which are protein function prediction, drug-target interaction prediction, binding affinity prediction and visualization of chemical space.

1.2 Contributions

In this thesis, several methods and tools were developed for drug-target interaction prediction and protein function prediction. Specifically, the major contributions can be explained as follows:

In Chapter 3 (DEEPred):

- A hierarchical prediction system was proposed which utilizes Gene Ontology directed acyclic graph structure to create stacked models for protein function prediction;
- Automated annotations of UniProt were included in the training set of the predictor, with the aim of enriching training data (especially for the GO terms with insufficient number of training instances).

In Chapter 4 (DEEPScreen):

- The idea of using compound images for predicting the interactions with target proteins and employing established convolutional neural network architectures;
- A reliable and open access reference dataset was created by filtering and preprocessing entire ChEMBL database whose aim is to use in DEEPScreen and other future studies;

In Chapter 5 (MDeePred):

- A novel protein representation method was proposed based on the amino acid pair properties in a protein sequence. Protein represented as multiple 2D feature channels where channels represent diverse properties of these amino

acid pairs. The proposed featurization approach can be applied to other fields of bioinformatics such as protein function prediction.

- A hybrid pairwise input neural network was created which extracts complex representations of compounds and proteins separately from individual inputs of them and then merges these complex representations to infer the binding relation of the input pair.

In Chapter 6 (iBioProVis):

- An interactive web-based tool was prepared for the researchers which allows investigation and analysis of how active compounds of different target proteins are distributed on a 2-D space and prediction of bioactivity profiles of new/existing/user-input compounds.

1.3 Achievements and Highlights

Achievements and highlights in this thesis are given below:

- DEEPred was one of the earliest applications of deep learning on protein function prediction area and it was among top five methods in CAFA PI protein function prediction challenge.
- Initial version of MDeePred performed fourth among 241 prediction submissions in *DREAM Drug-Kinase Binding Prediction Challenge.*
- Although there are several methods published in the literature in the last years, many of them do not provide open-access datasets and source codes for reproducibility. In all of our methods, source codes and datasets were made open to public for reproducibility and open-access tools were provided for researchers.
- For each method, thousands of models were trained and fine-tuned and made available for researchers.

- For each method, novel training/test/validation datasets were created for positive and negative datasets by carefully preprocessing and filtering corresponding databases.

1.4 Structure of the Thesis

This thesis was divided into six main chapters. The first chapter, introduction, defines the basic terminology, provide statistics regarding the relevant information stored in source biological and chemical databases, summarizes the experimental procedures along with computational approaches in drug discovery. The problem statement and the contributions are also given in the introduction part. In the second chapter, our protein function prediction method called DEEPred is explained in detail. The third chapter describes proposed DEEPScreen method for drug-target interaction prediction. In the fourth chapter, our binding affinity prediction method called MDeePred is explained. Fifth chapter demonstrates our bioactivity space visualization method called iBioProVis. Finally, discussion, conclusion and perspectives are given in the sixth chapter.

CHAPTER 2

LITERATURE REVIEW

[2]Virtual screening has the potential to greatly reduce the cost and time required for high-throughput screening [12]. Although the main purpose of virtual screening is to identify new drug candidates for specified targets, it also has other applications such as finding beneficial drug pairs [13] and the prediction of ATC codes for known drugs [14], [15]. In addition, the computational approaches mainly employed in virtual screening can also be used for *drug repurposing and off target effect identification*, where the aim is to find new uses for the already approved drugs [16]. Drug repurposing is an important research area since the approved drugs are already tested for safety issues; therefore, the cost and the required time for marketing repurposed drugs is much less than discovering and marketing novel drugs [17]. There are various examples of repurposed drugs in the market, most of which are being used for treatments of multiple diseases [18].

There have been several successful applications of virtual screening in detecting compounds with high affinities against pre-specified targets [19]. Some of these drug candidate compounds have also passed the clinical trials and became marketed drugs [20]–[24]. Doman *et al.* showed that their virtual screening approach substantially improved the rate of identified drug candidates against protein tyrosine phosphatase-1B (PTP1B) enzyme. The authors experimentally showed that the hit rate of their method was 34.8%, whereas the hit rate of the high-throughput screening experiment was only 0.021% [25]. Another successful application of virtual screening was

[2] Some parts of this chapter were published in *Briefings in Bioinformatics* journal in 2019 [7]. Please note that only the parts that I worked on were included from our publication.

proposed by Powers et al. which led to the discovery of a novel inhibitor of AmpC ß-Lactamase [26].

Both in high-throughput screening experiments and in conventional virtual screening approaches, the aim is to identify whether a given set of compounds are bound to a pre-specified target protein or not. In these applications, off-target effects are generally overlooked and other possible targets of the compounds cannot be identified. However, it is known that most of the bioactive compounds act on multiple targets (which causes these off target effects); in fact, the cases where a compound interacts with only a one target protein are considered as exceptional [27], [28]. The identification of the off-target effects is crucial to obtain potential side effect and toxicity information of the test compounds. For this purpose, another type of computational approach, target prediction (also known as the reverse virtual screening), was proposed [29], [30]. In target prediction, a compound is screened against a large set of proteins with the aim of identifying all possible targets of the corresponding compound (Figure 2.1). Generally speaking, the goal of both approaches is the prediction of unknown interactions between various compound-protein pairs.

In this literature review, the objective is to provide an overview of recent applications of computational drug discovery methods, called virtual screening, where the aim is to predict the bio-interactions between drug-like small molecules (i.e., compounds) and potential target proteins for the identification of novel drugs, using structural and physicochemical properties of compounds and targets along with the experimentally known (i.e., validated) bioactivities. Various data resources were explored that provide vast amount of information, which is essential for conducting virtual screening studies. Novel machine learning approaches were also investigated with recent applications to drug-target interaction prediction. In this framework, we discussed in detail the recent applications of deep learning techniques, which outperformed state-of-the-art virtual screening methods. Finally, we stated our observations and comments about the current status of the field of virtual screening.

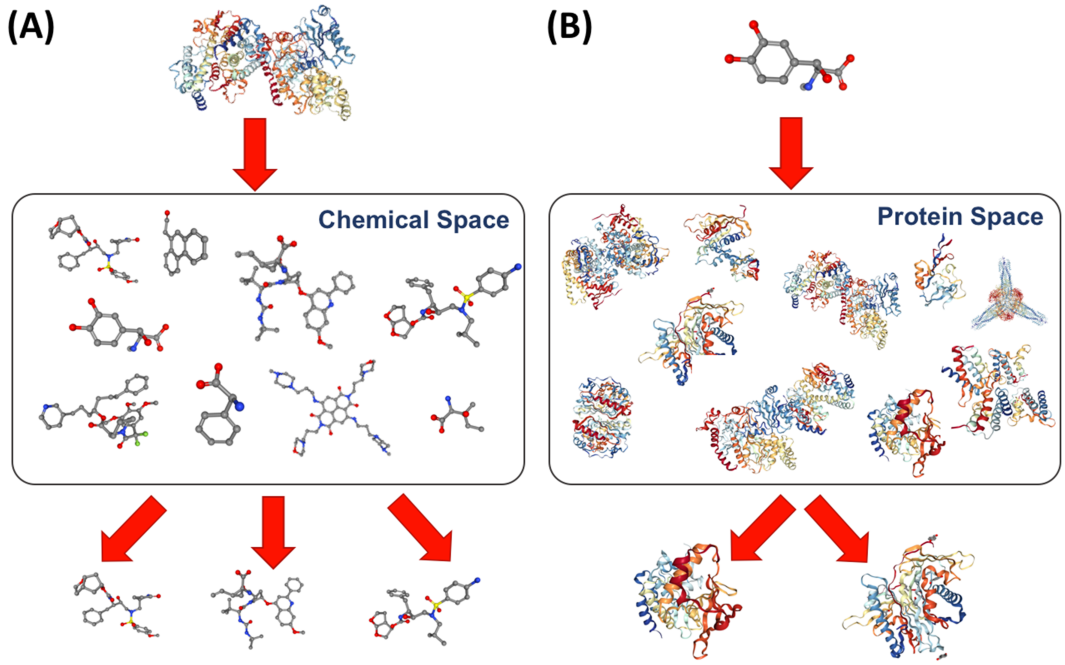


Figure 2.1. (A) In conventional virtual screening, multiple compounds are screened against a pre-specified target and candidate interacting compounds (i.e., ligands) are identified; whereas (B) in target prediction (i.e., reverse virtual screening), a compound is searched against multiple proteins and candidate targets are identified (Adapted from our publication [7]).

Most of the virtual screening methods make use of biological, topological and physicochemical properties of compounds and/or targets along with the experimentally validated bioactivity values of compound-target pairs to predict the unknown activities [31], [32]. For this, it is required to computationally record the compounds and targets as quantitative vectors (i.e., representations and descriptors) according to their molecular features. Virtual screening methods use these feature vectors as input in order to model the interactions between compounds and target molecules. VS methods can be divided into three groups based on the employed input features:

- *Structure-based virtual screening* employs 3D structure of targets and compounds to model the interactions [33], [34],

- *Ligand-based virtual screening* uses the molecular properties of compounds (mostly non-structural) in order to model the interactions with targets [31], [35], [36],
- *Proteochemometric modeling (PCM)* approach models the interactions by combining non-structural descriptors of both compounds and targets at the input level [37]–[40].

Previously, virtual screening was mainly divided in two groups (i.e., structure-based and ligand-based methods) [41], [42]; however, recent advances in PCM have put this field forward to be considered as a third group [39]. Both ligand-based and PCM methods can be considered as *non-structure-based virtual screening methods*. The field of ligand-based virtual screening have been extensively reviewed by Geppert *et al.*, and Lavecchia and Di Giovanni [35], [36]. In another study, Glaab reviewed the recent developments in both ligand and structure-based virtual screening approaches. The author defined a comprehensive pipeline for virtual screening over a target protein of interest and overviewed workflow management systems. The whole process was divided into four main steps such as data collection, pre-processing, screening, selectivity and ADMETox (i.e., absorption, distribution, metabolism, excretion and toxicity) filtering; and explained each step with a focus on relevant open-access software and databases. The author also implemented a downloadable cross-platform software by integrating open-access screening tools using the Docker platform [43]. Qiu *et al*. introduced the emergence of PCM and mentioned its advantages by referring to studies in which PCM models outperform conventional QSAR models in DTI modelling. The authors focused on the recent progress in PCM modelling in terms of target descriptors, cross-term descriptors and application scope of PCM including protein-small molecule and protein-macro molecule interactions. The authors reported that, with further advancements in molecular representations, machine learning techniques and the available bioactivity data, it may be possible to generate PCM models for more complicated systems such as ligand-catalyst-target reactions, which could provide help to identify biochemical

reactions more accurately [39]. The field of PCM was also reviewed by van Westen et al. and Cortés-Ciriano et al. [38], [40].

Structure-based virtual screening methods can only be applied when the 3D structure of both targets and the candidate compounds are available, which are either experimentally determined by X-ray crystallography or NMR, or predicted by computational approaches such as the homology modelling. Once the 3D structural information is obtained, docking can be applied to find interactions between a compound and a target, which predicts compound conformations in the binding site of the target using search algorithms and ranks them via scoring functions representing estimated binding affinities [25], [29]. Some of the most commonly used docking tools are AutoDock [44], DOCK [45], Glide [46], GOLD [47], FlexX [48] and Fred [49]. These methods rely on the conformations of atoms in 3D space; as a result, they are computationally intensive since the number of possible conformations of proteins and compounds increase exponentially with the increasing number of rotatable bonds. Moreover, the calculation of binding energies is a problematic issue [19]. In addition to these traditional methods, there are also similarity-based docking approaches such as HomDock [50], *e*SimDock [51] and *fkcombu* [52] that utilize structural similarities of compounds to predict their protein-bound states by aligning them on the experimentally determined 3D structure of a reference compound that is in complex with a target protein or evolutionarily related structures of that target protein [51]. Therefore, they do not require searching for low energy conformations of compounds contrary to conventional methods, which reduces the computational cost and makes them faster than traditional docking methods [50]. Both approaches can achieve high performance in estimating the interactions; however, their applicability is limited since the structural information is not available for the majority of the proteins and compounds, and the experimental identification of the 3D structures is challenging [9]. Although homology models of proteins can be used as templates for docking, it is not possible to obtain a reliable model for all proteins due to lack of a reference protein structure that is evolutionarily close to the target protein to be modelled. Even if similarity-based docking

approaches are less sensitive to weakly homologous protein models [51], they are not feasible in the absence of similar compounds to the reference compound. Therefore, non-structure-based virtual screening methods are more preferable if a reliable target structure is not available [53]. It was reported in the literature that the non-structure-based methods have a similar potential to detect drug targets as the structure-based methods [54]. In addition, several studies showed that structure and non-structure-based methods often provide complementary results [30], [55]–[57]. There are also hybrid-type methods that combine 3D structure information along with the ligand-based information in the literature [53]. Structure-based virtual screening methods are out of the scope of this study and information about this field can be obtained from the literature [33], [34], [54], [58], [59].

2.1 Machine Learning Applications in Virtual Screening

The field of machine learning has been extensively reviewed and discussed in several books [60]. There are two main approaches in machine learning literature in terms of how the learning process is carried out, *supervised learning* and *unsupervised learning*. In supervised learning, the objective is to infer a function that maps the input data to the output class labels [61]. Whereas the aim in unsupervised learning is to learn the hidden structure of input data without having class labels. Unsupervised learning algorithms employ techniques to discover relationships among the non-labeled input samples. The most popular applications of unsupervised learning are clustering and dimensionality reduction. Once the groups and clusters are obtained with the application of an unsupervised learning method, each group can be inspected to assign semantic meanings by experts [62]. Both supervised and unsupervised machine learning techniques are used in cheminformatics on a wide range of topics including virtual screening [63], [64], [73], [74], [65]–[72] yet most of the methods so far assumed the supervised approach. A plethora of methods have been proposed for virtual screening purposes in the last decade. These virtual screening methods use experimentally validated

compound-target pairs and their features along with the bioactivity information to create predictive models for future predictions of activities.

In terms of the methodological utilization of the input properties, virtual screening methods can be divided into similarity-based and feature-based methods, although there is no such technical classification in the machine learning literature [60], [61], [67], [75]. In the following sections, similarity-based and feature-based virtual screening methods are investigated, which is followed by the recently popularized deep learning based applications in virtual screening. For this, we have mostly focused on the studies published in the last five years, some of which have aims beyond DTI prediction (e.g., estimation of beneficial drug-drug combinations or ATC code prediction).

2.1.1 Similarity-based Approach

Similarity-based methods rely on the assumption that biologically, topologically and chemically similar compounds have similar functions and bioactivities and therefore they have similar targets [73], [76]–[78]. In the similarity-based approach, the target associations of similar compounds (or the compound associations of similar target proteins) are transferred between each other. Therefore, *transfer approach* is a term used interchangeably to define similarity-based methods. In chemical space, similarities are calculated by searching molecular substructure and isomorphism based on the representations of molecules such as SMILES and InChI. In target space, similarities are mainly calculated by sequence alignment methods. The methods under this approach construct similarity matrices either for compounds or targets, or for both them [75]. Subsequently, constructed similarity matrices are used by the machine learning models. Below, we provided reviews for three similarity-based VS methods, which were published in the last few years.

With the aim of identifying biologically and structurally similar clusters of compounds, weighted clustering was proposed by integrating multiple similarity

matrices [73]. Two datasets were used: the epidermal (EGFR) and the fibroblast growth factor receptor (FGFR) datasets. EGFR dataset contained bioactivity assay readouts and gene expression profiles for 35 compounds and 3595 genes. In FGFR dataset, the chemical structure information, gene expression data and bioactivity assay readouts were available for 94 compounds and 1056 genes. Two similarity matrices were generated based on the structural and the phenotypic properties. Structural properties of compounds were represented by ECFP6 fingerprints and similarities of compounds were calculated using the Tanimoto coefficient. For the phenotypic similarity matrix calculation, bioactivity readouts were used. The Euclidean distance was employed to calculate the phenotypic similarities between two compounds based on their bioactivity results on the same assays. Subsequently, generated similarity matrices were used to perform clustering using a weighted clustering algorithm. The weighted clustering technique was shown to be more efficient in terms of identifying structurally and biologically similar compounds compared to the individual clustering methods.

A supervised similarity based PCM method was described for the detection of: *(i)* interactions between new drug candidates and known targets and *(ii)* interactions between new drug candidates and new targets [77]. The similarity between two compounds was measured by a combination of non-structure-based score (ATC-based semantic similarity score) and 2D graph structure-based score. ATC-based semantic similarity score was calculated by counting the common subgroups between ATC code annotations of two compounds. 2D structure-based similarity calculation was performed by aligning graph structures of compounds. The similarity score for a pair of targets was computed using a combination of a functional-similarity-based (using Enzyme Commission -EC- numbers) score and a sequence-based similarity score. Functional similarity-based score was calculated by counting the number of common EC number annotations. For sequence-based similarity score calculation, subsequences in the ligand-binding domains were extracted and they aligned the extracted subsequences to calculate similarity scores between targets. The datasets constructed by Yamanishi *et al.* [79] for four classes of targets (i.e.,

GPCRs, ion channels, enzymes and nuclear receptors) were employed for the tests. A concept called "super-target" was proposed to overcome the problem of the scarcity of training instances in terms of targets. Similar targets were clustered and it was assumed that if the drug interacted with a target, it would also interact with the other targets in the same super-target cluster. For the prediction of new drug candidates for a known target, the following methodology was pursued: When a new compound was given as input to the system, for each known target t_x, a confidence score was calculated between the query compound and the super-target cluster that t_x belonged to, based on the drug associations of the targets in that super-target cluster. Subsequently, another confidence score was calculated between query compound and t_x based only on the drug associations of t_x. Finally, these two scores were combined as a single prediction score. For the prediction of new drug candidates for a new target, a similar procedure was followed. In this case, the new target was considered as a member of most similar super-target cluster based on its functional and sequence similarities.

SwissTargetPrediction is a supervised similarity-based method that combines 2D similarity and 3D similarity of compounds with the aim of identifying new targets for query compounds [76]. ChEMBL database was employed to obtain known compound–target pairs. The training dataset consisted of 280,381 small compounds for 2,686 targets. When a compound was given as input to the system, a combination of 2D and 3D similarity scores were calculated between the query compound and all compounds with known targets. In order to obtain 2D similarity score, a compound was represented by FP2 fingerprints and the 2D similarity scores between the query compound and other compounds were calculated by the Tanimoto coefficient. For the 3D similarity score, 20 different conformations of compounds were generated and the Manhattan distance was used to calculate distances among all conformations of two compounds. The smallest distance was then chosen among the 20x20 distance scores and it was converted into a 3D similarity score. 2D and 3D similarity scores were combined as a single prediction score for targets. Finally, the system outputs a ranked list of targets based on the combined similarity scores. Users can get

predictions for a compound using SMILES string of the query compound or by drawing 2D structure of compounds using the web tool provided. SwissTargetPrediction is available at http://www.swisstargetprediction.ch.

2.1.2 Feature-based Approach

In *Feature-based virtual screening* methods each instance (i.e., compound and/or target) is represented by a numerical feature vector, which reflects various types of physicochemical and molecular properties of the corresponding molecules. Targets are usually modelled using their physical and chemical properties, subsequence distributions or functional attributes; whereas the compounds are usually modelled using structural properties. In a typical feature-based virtual screening application, a set of compounds that is known to interact with a specific target is extracted from compound and bioactivity databases. Subsequently, feature vectors are generated for each compound. Finally, the constructed feature vectors are fed to a machine learning algorithm to create a predictive model for the interaction with the corresponding target. When a new query compound's feature vector is given to the trained model as input, the output of the predictive model is either active or inactive against the corresponding target protein (Figure 2.2). This is the so-called *ligand-based* approach in terms of the incorporated input feature types (i.e., compound features). PCM methods also assume a similar methodology, but they jointly model the target properties at the input level along with the compounds, so that the query can be a compound-protein pair, and the model predicts the presence of that specific interaction. Examples of feature-based virtual screening methods are given below.

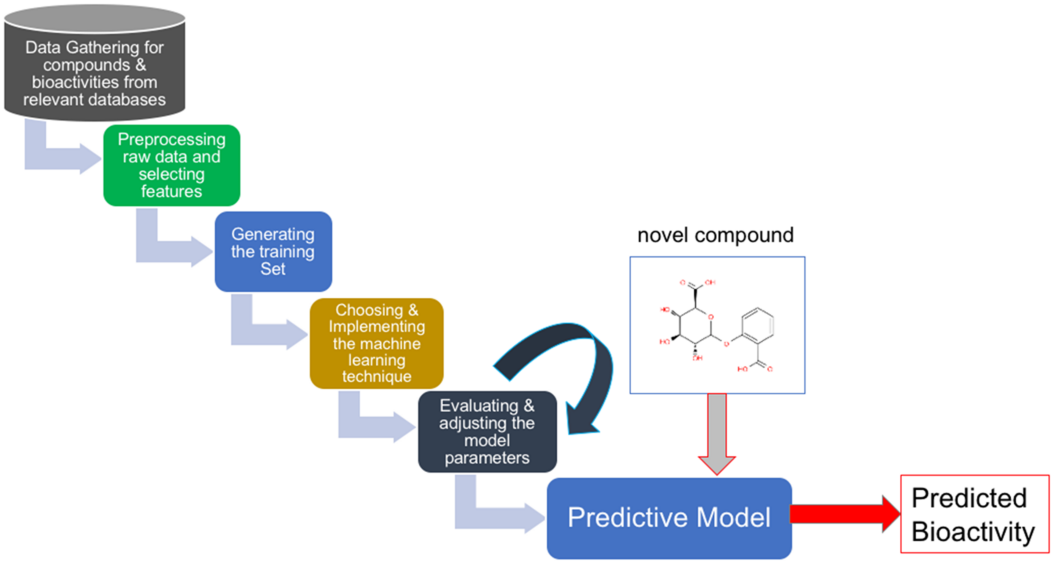


Figure 2.2. The steps of a typical feature-based virtual screening method for training a predictive model (Adapted from our publication [7]).

A supervised machine learning methodology was proposed by Liu *et al.* [14] using a combination of both similarity and feature-based approaches to predict drug–ATC code associations. DrugBank database was employed to create their positive and negative training datasets. The total set was composed of 1333 small molecule drugs and their ATC codes at various levels. ATC code prediction problem was described as a binary classification problem. Therefore, for each ATC code a positive training dataset and a negative training dataset were constructed. Known drug-ATC code pairs were retrieved to construct the positive training datasets. To construct a negative training dataset for each ATC code, they first removed the positive drug-ATC code pairs from all possible drug-ATC code pairs and randomly selected samples from the remaining set. Then six scores were defined to calculate drug–drug similarities, which are based on chemical structures, functional groups, target proteins, drug-induced gene expression profiles, side-effects and chemical–chemical associations. Each drug was represented as a 6-dimensional feature vector. The value of a certain feature was determined by taking the largest similarity score between the input drug and the drugs associated with the corresponding ATC code. Once the drugs were converted into feature vectors, the logistic regression method

was used to train predictive models for each ATC code. When a new query compound is given to the system, first it is converted to the feature vector based on the similarity values; then, it is given to the predictive models as input to predict the candidate ATC codes. The method, SPACE, is available at http://www.bprc.ac.cn/space.

In the work by Cano *et al.* [80], the main objective was the inherent selection/ranking of features and training a DTI prediction classifier using random forest. Directory of Useful Decoys (DUD) was used to create their training dataset, which was composed of kinases, nuclear hormone receptors and other proteins. The constitutional, charged partial surface area and fingerprint-based descriptors were the input to the system. The performance of the model was compared with SVM and neural network classifier based models and the random forest classifier was successful to select and rank most representative features given a large set of input features. In this setting, it was also observed that a reduced number of features drastically decreased the computational complexity of DTI prediction models.

For drug repurposing, a combination of similarity and feature-based supervised method was proposed by integrating drug/compound, target protein, phenotypic effect and disease association data from several sources [56]. The chemical structures of drugs and compounds were retrieved from the ChEMBL database. Three different molecular descriptors were used to represent compounds, which are Extended Connectivity Fingerprints (ECFP4), Chemistry Development Kit (CDK) Fingerprints, and KEGG Chemical Function and Sub-structures (KCF-S). The compounds were thus represented by 1024, 1024 and 475,692 dimensional fingerprints. The obtained feature vectors were referred as the "chemical profile" of the compounds. Phenotypic effects of drugs were obtained from Food and Drug Administration (FDA) Adverse Event Reporting System (FAERS) and each of the 2,594 drugs were represented as a 16,075-dimensional feature vector, where each dimension represents the presence or absence of a phenotypic effect. This dataset was named as the "phenotypic profile" of a drug. Compound-target interactions and the bioactivity values were obtained from seven different databases. Their total

activity set was comprised of 1,287,404 interactions involving 519,061 compounds and 3,736 targets. This dataset was referred as the "chemical protein interactome dataset". Molecular features of diseases were obtained from the International Classification of Diseases (ICD10) and the KEGG DISEASE database. The diseases were represented as 6342-dimensional binary feature vectors, where each dimension represents presence or absence of a molecular feature. Drug-disease associations were obtained from medical books and from the KEGG DRUG database. This dataset was comprised of 5,830 drug-disease associations involving 2,271 drugs and 463 diseases. Disease-target associations were obtained from the KEGG DRUG database. They created a dataset consisting of 2,062 disease-target associations for 250 diseases and 462 therapeutic target proteins and this dataset was named as the "disease-target association template". Their prediction method was composed of three parts, which were called as the Target Estimation with Similarity Search (TESS), Indication Prediction by Template Matching (IPTM) and Indication Prediction by Supervised Classification (IPSC). In TESS, the aim was to predict potential targets of a given drug based on similarity search. Each compound was represented by a 3,736-dimensional target interaction profile. The similarity search was performed against the compounds in the chemical-protein interactome dataset based on the chemical and phenotypic profiles of the compounds. Subsequently, for each target, the compounds that were associated with the corresponding target were retrieved and the drug-target similarity score was assigned using the similarity score between query drug and the most similar compound that were associated with the corresponding target. In IPTM, the aim was to predict novel drug indications for the query drugs. First, target proteins of the query drug were retrieved. For each target, the diseases that were associated with the corresponding target were obtained from the disease target association template. This way, the query drug was linked to the diseases based on their target associations. In IPSC, the aim was to predict novel drug indications using a supervised classification method. In this method, target proteins of the query drug and molecular features of diseases were used. Each drug-disease pair was represented by a feature vector and drug indication prediction was

formulized as a binary classification problem, where the output of the regression-based classifier shows if the drug could be applicable to the paired disease. The cross-validation results showed that IPCM and IPSC methods outperformed the previous methods from the literature.

A supervised feature-based PCM method was proposed for GPCR and protein kinase targets [81]. The positive training dataset was generated using the GLIDA database by extracting experimental compound-target interactions, containing 5,207 interactions for 317 targets and 866 compounds [82]. Negative training samples were generated among the unknown compounds-target pairs. Compounds were converted into 929-dimensional molecular descriptors. Descriptors for targets were generated using a string kernel, resulting in 400-dimensional feature vectors. Finally, two vectors, that is, compound and target descriptors, were concatenated for each positive and negative interaction. Finally, the generated feature vectors were fed to an SVM classifier to train predictive models for each target family. Selected novel drug predictions were also experimentally validated for both GPCR and protein kinase families.

A supervised feature-based PCM method for the identification of novel drug combinations was described by Iwata *et al.* [13]. Orange Book and the KEGG databases were proposed to extract beneficial drug-drug combinations [83], [84]. Interacting drug–target pairs were collected from seven different databases. 4,007 drug-target interactions were incorporated for 588 drugs and 930 targets. Each drug was represented by a 1,078-dimensional binary feature vector where 930 of them represent the presence or absence of each target and 148 of them represent the presence or absence of ATC code annotations. Subsequently, each drug–drug pair was represented as a binary feature vector by combining individual feature vectors of the corresponding drug pairs. Finally, the obtained feature vectors were fed to a logistic regression classifier. When a new drug–drug pair is given as a query to the system, the output was calculated as potentially beneficial or not.

Another supervised PCM method was proposed for drug-target interaction prediction [85]. In this approach, compounds were represented using fingerprints and targets were expressed as sequence alignment based profiles. First, the position specific scoring matrices were generated for all target protein sequences. Subsequently, local binary pattern method was adapted to extract features from position specific scoring matrices. In the end, targets and compounds were represented by 256 and 615-dimensional feature vectors. Next, principal component analysis was applied for both target and compound feature vectors to obtain an uncorrelated and a reduced number of features. Four different datasets were employed: enzymes, GPCRs, ion channels and nuclear receptors. The positive samples were interacting compounds-target pairs and same number of negative samples were selected randomly from remaining interaction sets. Finally, obtained features were fed into discriminative vector machine classifier which was proposed by the same group. Support vector machine classifier based on the same features were trained and the performance of two classification methods was compared. The results were compared with three conventional methods and this method had a better performance.

In terms of the methodological approach used in modelling the pairwise relationships, a highly studied topic is the development of network or graph analysis based DTI prediction methods. In these methods, compounds and targets are represented as nodes on a graph, where the edges connecting these nodes indicate interactions. Modelled this way, estimation of unknown DTIs becomes a link prediction task. Various techniques, borrowed from the fields of graph theory and social and biological network analysis, are employed to solve the problem at hand. Frequently, the relationships in-between the compounds (i.e., in terms of molecular/structural similarities) and in-between the targets (i.e., in terms of homology or protein-protein interactions) have been incorporated in the generated networks to enrich the input information. An advantage of the network/graph-based approach is that the system can work well even when the number of training instances is very low. Network/graph-based DTI prediction methods can be similarity-based, feature-based or a combination of both. One seminal work on this

subject is by Yamanishi *et al.* [86], where the authors integrated both the similarities within the genomic space (using pairwise sequence alignment) and within the chemical space (using molecular and pharmacological effect similarities) in their network, for the first time. In this study, chemical, pharmacological and genomic spaces are unified and used together with the known DTIs to generate predictions for target families of enzymes, ion channels, GPCRs and nuclear receptors. In another study, Gönen [87] incorporated target protein sequence similarities and compound molecular structure similarities in a kernelized Bayesian matrix factorization framework to predict unknown DTIs. Other examples for network/graph based methods can be given as Shi *et al.* [77], Sawada *et al.* [56] and Li *et al.* [85], which are reviewed in this study. It is also important to note that the gold-standard dataset generated by Yamanishi *et al.* (explained in the section entitled: "Gold Standard Datasets for Virtual Screening") is suitable for testing network/graph-based DTI prediction methods.

In a review study by Chen *et al.* the available resources for DTI prediction were presented including databases, web servers and computational methods [88]. Methodological approaches were categorized as graph/network-based, machine learning-based and other methods, and the advantages and disadvantages of each approach were discussed. For graph/network-based drug discovery, the integration of different network models and sequencing technologies have been indicated to provide significant improvements for personalized medicine. As a suggestion to further improve the DTI prediction performance, the employment of heterogeneous training data by combining different data sources was recommended. The graph/network-based approach (excluding artificial neural networks), which was highly employed in the DTI studies especially between 2006 and 2013 [79], [86], [87], [89]–[94], was mostly left out of this study in order to focus on novel DTI prediction approaches.

Both the similarity and the feature-based approaches are used extensively in the literature. One of the main advantages of similarity-based approach is that when the problem involves heterogeneous data, different types of similarity matrices can be

combined in the same model. Another advantage of similarity-based methods is that, sophisticated kernel methods can be applied [75]. They are also relatively simple and easy to model. However, they are computationally not practical to apply on large datasets since they require extremely high number of similarity calculation operations. Considering the feature-based methods, one advantage is that, they can reveal intrinsic properties of compounds and targets that play critical roles in DTIs, which leads to more interpretable results. Another advantage is that, a problem-specific feature selection can be performed to obtain relatively more accurate predictions. One of the challenges about the feature-based methods is the selection of negative samples for the construction of negative training sets. Although chemical databases include experimentally validated drug-target interactions, they do not provide sufficient number of experimentally validated non-interacting compound–target pairs. When this is the case, the frequently employed approach for negative sample selection is to randomly select pairs from the set remained after excluding the positive training samples. However, this approach is problematic since the randomly selected pairs may also include pairs that are interacting, which is unknown (and therefore not recorded in the source database) at the time being. Negative sample selection is not only a problem for the virtual screening field, but also a problem for cheminformatics and bioinformatics in general [95]–[97]. There are alternative algorithmic methods to construct more reliable negative training datasets [75], [88], [98]–[100]. The lack of sufficient negative training datasets also leads to the class imbalance problem, which highly effects the prediction performances of computational systems. The class imbalance problem may produce a bias towards the class having most training samples, causing the model to give excessive number predictions for this class, resulting in high number of false positive predictions. In their recent studies, Soufan *et al.* focused on the class imbalance and false positive prediction problems. Models using five different solutions were trained to overcome class imbalance problem and the performances of these systems were compared. Classifier performance aware methods were also used along with several evaluation metrics in order to reduce the false positive rates [101], [102]. Another

challenge for the feature-based methods is the high-dimensionality of feature vectors, which can reach the order of millions [92], [103]. Extremely high-dimensional vectors create computational overhead, and they often lower the accuracy of predictions. Usually combining different types of informative features increases the performance of classifiers; however, after a certain point, adding more features to the system starts to decrease the performance, which is known as *curse of dimensionality* [104]. Therefore, feature-based methods may require the application of feature selection techniques to reduce dimensions and to keep only the most relevant and distinctive features in the model. Various studies have been performed to analyse and compare feature reduction and selection techniques in the literature [105]–[110].

2.1.3 Deep Learning Applications in Virtual Screening

Deep learning algorithms have been widely used in recent years due to their successful results in computer vision, speech recognition and bioinformatics [111]–[114]. The term deep learning represents a group of machine learning approaches, which contain multiple data processing layers. Deep learning algorithms yield successful learning of the representations of the input data through multiple levels of abstraction [115]. Deep neural networks (DNNs) are artificial neural network methods that have multiple hidden layers. In this sense, DNNs are considered as a group of deep learning algorithms. DNNs convert the low-level features obtained from the input into more and more complex features in each subsequent layer. An example of a basic feed-forward DNN (i.e., a multilayer perceptron - MLP) architecture is given in Figure 2.3, along with other popular DNN architectures. In this figure, nodes correspond to neurons and the edges between nodes correspond to neural connections, where the signal is transmitted. According to the model choice, neurons at different layers can be fully-connected to each other or not. At each neuron, a non-linear activation function, whose coefficients are determined during the training procedure, takes the input signal from multiple connected neurons at the

preceding layer and modifies it before transmitting it to the next neuron. A standard feed-forward artificial deep neural network has three different types of layers: the input layer, hidden layers and the output layer, each of which are composed of multiple parallel-connected neurons. A neural network with two or more hidden layers is considered as a deep neural network [111] The input features are directly fed to the input layer and after a number of non-linear transformations using hidden layers the predictions are generated at the output layer. Each output node corresponds to a task (i.e., class) to be predicted. If there is only one node in the output layer then the corresponding network is referred as a single-task DNN. Otherwise, it is called a multi-task DNN.

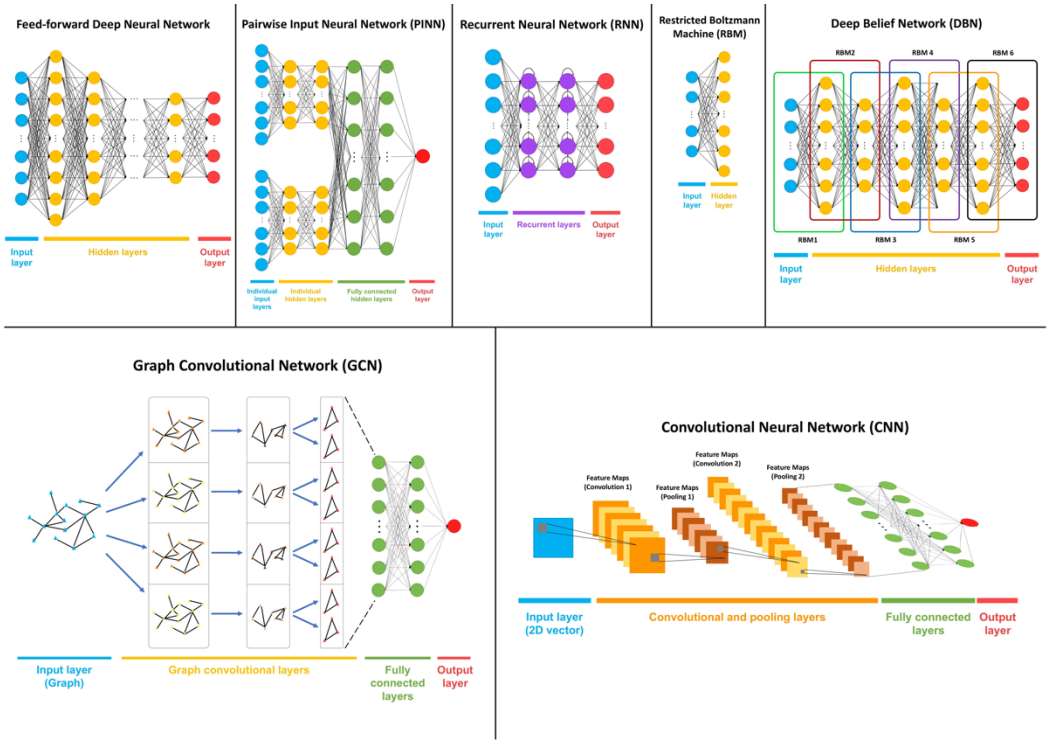


Figure 2.3. Schematic representations of different deep neural network (DNN) architectures frequently used in the literature (Adapted from our publication [7]).

A deep learning algorithm won the Kaggle Virtual Screening Challenge, which was sponsored by Merck, and it drew considerable attention to employing deep learning techniques for virtual screening purposes [116], [117]. Recently, it was shown that

deep learning algorithms outperformed the state-of-the-art methods in numerous virtual screening studies [63], [116], [118]–[124]. Several advantages of deep learning architectures have been reported for virtual screening:

- deep learning algorithms inherently build relationships between multiple targets, therefore they are suitable for multi-task learning;
- they provide higher level abstractions by building complex features from raw input data in a hierarchical manner and able to identify the unknown structure in the data, and the observed high performance of DNNs is usually attributed to this ability;
- shared hidden units among the targets enhances the prediction results of the targets having less training samples.

There are several DNN techniques (or architectures) and each has advantages and disadvantages according to the nature of the data being analyzed and the types of features employed. The most commonly used ones can be listed as feed-forward DNNs with multiple hidden layers [117] which can be considered as the standard application, deep convolutional neural networks -CNNs- (highly used in computer vision) where each of the several convolutional layers will capture a specific feature from the multi-structured input data [118], [123], and pairwise input neural networks (PINN) where the features belonging to compounds and proteins can be fed to the model together [125]. DNN-based techniques are also divided into two according to the number of prediction tasks in a model, such as the single-task and multi-task DNNs. Single task networks are modelled in such a way that one model can only produce answer for one specific question (e.g., is there an interaction between this compound-protein pair) [126]; whereas, multitask networks are modelled to infer multiple unknowns in one model (e.g., which of the 20 potential target proteins can interact with the input compound) [116]. All of these DNN architectures can be considered under the title of feature-based machine learning methods. Below we review a large collection of studies of deep learning applications in computational drug discovery with an emphasis on DTI prediction. Table 2.1 summarizes

frequently used DNN architectures in the field of virtual screening and groups the reviewed studies in terms of the employed DNN architectures. Figure 2.3 shows the schematic representations of those DNN architectures explained in Table 2.1.

Table 2.1 Deep learning architectures together with the virtual screening studies that utilize each architecture.

Architecture Name	**Description**	**DNN-based Virtual Screening Studies**		
		Citation	**Input Protein Features**	**Input Compound Features**
Feed-forward Deep Neural Network – FFDNN (an interchangeably used term in some of the resources: multi-layer perceptron - MLP)	A feed forward deep neural network can be considered as the most basic DNN architecture, which have multiple hidden layers that are usually fully-connected to each other (Figure 5). These networks are mostly structured to predict multiple number of tasks (usually targets in DTI prediction) in a single model (i.e. multi-task networks).	Dahl *et al.* [116]	-	Several different molecular descriptors
		Ma *et. al.* [117]	-	Atom pairs and donor-acceptor pair descriptors
		Unterthiner *et al.* [124]	-	ECFP12
		Ramsundar *et al.* [121]	-	ECFP4
		Koutsoukas *et al.* [126]	-	ECFP4
Pairwise Input Neural Network (PINN)	PINNs are feed-forward neural networks that take two different feature vectors as input and predicts their relation as output. In some of the PINN applications, the two individual input vectors are processed by separate groups of neurons before they are merged at a subsequent fully-connected layer. PINNs are especially suitable for the prediction of pairwise relations such as DTIs.	Wang *et al.* [125]	Binding sites	2D structural fingerprints
		Wan *et al.* [127]	Amino acid triplets in protein sequences	2D structural fingerprints
		Lenselink *et al.* [128]	Physicochemical properties	Morgan fingerprints

Table 2.1 (continued)

Recurrent Neural Network (RNN)	RNNs are specialized artificial neural networks that contain feedback loops to extract patterns using not only the current input but also the previously perceived inputs. RNNs successfully extract patterns from sequential data such as texts, protein sequences, audio signals and time series data. RNNs mainly have applications in speech recognition.	Goh *et al.* [129]	-	SMILES strings
Restricted Boltzmann Machine (RBM) / Deep Belief Network (DBN)	RBMs are single layer generative artificial neural networks, which can learn probability distributions given the training data. Deep Belief Networks (DBN) are constructed by stacking RBMs to solve more complex problems. Different from FFDNNs, DBNs are trained stack-by-stack. DBNs are used in several applications such as clustering and generating objects such as images.	Wen *et al.* [130]	Sequence composition descriptors	2D structural fingerprints
		Wang *et al.* [131]	Direct (e.g. compound-target binding) and indirect (e.g. compound changes the level of expression of the target) interactions on the multi-dimensional DTI network	
Convolutional Neural Network (CNN)	CNNs inherently extract the features hidden in the input samples by applying sequential layers of convolutions and pooling modules. The convolution layers extract local patterns (sub-features) by moving a window over the sample and the pooling layers are used to sub-sample and reduce the features. CNNs are mainly used in image processing applications.	Wallach *et al.* [123]	3D binding sites	3D structures of compounds
		Gonczarek *et al.* [132]	Binding pockets	3D structural fingerprints
		Goh *et al.* [133]	-	2D structure images of compounds
Graph Convolutional Neural Network (GCN)	GCNs are created by applying convoluting operations on graph encodings. GCNs can be used to model any entity that is expressed as a graph such as social networks and chemical compounds.	Kearnes *et al.* [134]	-	2D graphs of compounds
		Altae-Tran *et al.* [135]	-	2D graphs of compounds

One of the early studies employed multi-task feed-forward DNNs for the prediction of activities of compounds against 19 target assays from the PubChem database [116]. Active and inactive labels of compounds were used against each of the 19 targets and the training dataset was comprised of 69,396 active and 70,331 inactive compounds. The problem was stated as a classification problem, where inputs were the compound descriptors and outputs were the presence of interaction against the modelled targets. 3,764 dimensional molecular descriptors were generated to represent the compounds. The performance of multi-task neural network was compared with random forests, gradient boosted decision tree ensembles and logistic regression methods. The results showed that multi-task neural networks performed best in most of the cases. The performance of single-task and multi-task neural networks were compared as well and multi-task neural networks achieved better performance in the test cases. Feature selection was further performed However, no significant performance gain due to feature selection was observed.

In order to select hyper-parameters and compare single-task and multi-task DNNs, Ma *et al*. [117] made use of Merck's Kaggle challenge dataset along with the Merck's in-house datasets. Each compound was represented as molecular descriptors based on atom pairs and donor-acceptor pair descriptors. In total, there was 30 datasets, which included 129,295 unique compounds. Several models were created using different hyper-parameters and it was reported that the use of a single set of hyper-parameters can perform better than using optimized parameters for different datasets. The performance was compared with the performances of models trained with random forest classifier and DNNs achieved higher performance. Furthermore, on the average, multi-task DNNs obtained better prediction performance than the single-task DNNs. The performance of the single-task DNNs was reported to increase with the increasing size of training datasets.

In another early study, Unterthiner *et al*. [124] used multi-task DNNs for the prediction of activities of compounds for targets. ChEMBL database was used to obtain known compound-target interactions and the corresponding bioactivity values, which were discretized as active, weakly active, weakly inactive and inactive

based on pre-defined bioactivity thresholds. This way, a dataset was generated that comprised of 2,103,018 (972,268 active - 1,130,750 inactive) bioactivity measurements distributed across 5,069 targets and 743,336 compounds. In the models, each compound was represented as about 13 million dimensional fingerprints using ECFP12 features and then number of features were reduced to 43,340 dimensions by discarding the features that were absent in the majority of compounds. Finally, multi-task DNNs were trained where the inputs were compound feature vectors and the outputs were target activity values. The performance of their multi-task neural network was compared with support vector machine, binary kernel discrimination, logistic regression, *k*-nearest neighbour and Parzen-Rosenblatt methods. Multi-task neural network outperformed all other algorithms.

A particular type of DNNs, pyramidal multi-task DNNs was described and applied for virtual screening [121]. In this pyramidal architecture layers are organized such that each layer has less number of neurons than its previous layer Training datasets were collected from four different publicly available data sources, which consisted of nearly 37.8 million experimental compound-protein interactions for 1.6 million compounds and 259 targets. The compounds were represented by ECFP4 fingerprints. Several experiments were conducted by changing the number of tasks and training samples in their models. The performance of pyramidal multi-task neural networks was compared with logistic regression, random forest, single-task neural network, pyramidal single-task neural network and 1-hidden layer multi-task neural network. Pyramidal multi-task neural network performed best among the other methods. The following important observations were reported:

- the multi-task deep architecture achieved significant improvement over standard machine learning algorithms;
- the performance of multi-task networks increased as more tasks and data points were added;
- shared bioactive compounds among targets had a significant positive impact on performance.

The main difference between the study by Ramsundar *et al.* and the study by Unterthiner *et al*. is that, the number of known ligands for each target was much higher in this study (i.e., ~2 million samples for 1,230 targets vs. ~40 million samples for 259 targets). In addition, the main concern of the study by Ramsundar *et al.* was to discover the causes of performance changes based on parameter selections (i.e., number of tasks, training data sizes and layer organizations), whereas in Unterthiner *et al*. the main aim was to demonstrate the performance gain of multi-task DNNs over other baseline methods.

An investigative study was performed for virtual screening by Koutsoukas *et al*. [126] using single-task feed-forward DNNs. Their study was composed of two major parts: first of all, the effects of the hyper-parameter choices on the performance were investigated. In the second part, the aim was to compare the DNNs with other types of classifiers in terms of performance. ChEMBL database was used to create training datasets for seven different targets from diverse protein families and an individual prediction model was constructed for each target. 7,218 active compounds was tested against these targets and the compounds were represented as 1024-dimensional molecular fingerprints. The rectified linear unit activation function performed better than the other activation functions during the experiments. It was also reported that the number of neurons at each layer that give the best performance was highly dependent on the dataset and should be determined separately for each model. The drop-out regularization helped to gain better performances around 50% drop-out rate. In the second part of the study, the performance of DNNs was compared with Bernoulli Naive Bayes, k-nearest neighbour, random forest and SVM classifiers, and DNNs outperformed all of them.

Pairwise input neural networks where inputs represented pairs of target-ligand feature vectors are also a popular type of DNNs. Pursuing a PCM approach, Wang *et al.* considered target-ligand interaction as a binary classification problem, where inputs represented pairs of target-ligand feature vectors and the binary output represented the interaction prediction for the corresponding pair [125]. The training dataset was obtained from sc-PDB database and comprised of 836 targets, 2710

ligands and 6830 target-ligand pairs [136]. Binding sites of proteins were used as target features, which were represented as 199-dimensional vectors. The compounds were represented as 413 dimensional fingerprints. Subsequently, each known interacting target and ligand pair was labelled as a positive example and the remaining pairs were considered as the negative examples. This information was used then to train a four-layered pairwise neural network model. The method achieved better performance than the conventional methods from the literature in terms of several criteria.

Wan *et al.* [127] proposed a DNN for DTI prediction. Their framework also included an unsupervised representation learning for feature generation by identifying low-dimensional representations of the initial input features. The initial input features were composed of Morgan fingerprints for compounds and protein sequences for targets, which were embedded to a fixed low dimensional space (i.e., 200 dimensions for compounds and 100 for proteins) using natural language processing (NLP) techniques (i.e., latent semantic analysis and Word2vec). Sub-structures in compounds and amino acid triplets in proteins were treated as words for the embeddings. The system was trained on large-scale ChEMBL bioactivity data by generating training set sizes of 360,835 positive and 93,903 negative examples. These examples were selected using activity measurement values (i.e., IC50/Ki values $\leq$ 1 μM for positive and $\geq$ 30 μM for negatives). The performance was measured using *k*-fold cross-validation in different settings and it was compared against random forest as the baseline classifier, where the proposed approach significantly surpassed on the difficult-to-predict setting. The prediction performance was also measured on a test set composed of DUD-E gold-standard dataset interactions and compared to another deep learning based DTI prediction method AtomNet [123]. The elevated performance has indicated effectiveness of the of the word-embedding approach.

Lenselink *et al*. [128] proposed a PCM deep learning solution to drug-target interaction prediction. The training dataset was generated using verified bioactivities in the ChEMBL database. Target protein sequences were represented as 169-

dimensional feature vectors based on their physicochemical properties. Compounds were represented by varying lengths of Morgan fingerprints (e.g., 4096, 2048, 512, 256-dimensional). The interacting target-compound pairs were fed to multi-task DNN to create the predictive models. The performance change was investigated based on multiple criteria such as the length of the fingerprints, input feature utilization approach (i.e., ligand-based against PCM), the depth and the architecture of the DNNs. The performance was compared with the models trained by naive Bayes, random forest, support vector machines and logistic regression classifiers for both ligand-based and PCM approaches, where possible. As a result, the DNN models outperformed the models generated using conventional techniques and the average performance of PCM-based models was slightly higher compared to the ligand-based ones.

SMILES2vec is a RNN deep learning solution to predict the same physical properties of compounds directly using the SMILES representations as the input [129]. The aim here was also similar to their previous study in terms of performing minimal amount of feature engineering and pre-processing for model construction. Recurrent DNNs were used to train the predictive models and Bayesian optimization technique was used to select the best hyperparameters. The performance results of SMILES2vec was compared with the performances of DNNs trained using engineered features. According to the results, SMILES2vec outperformed other methods on regression tasks and underperformed on classification tasks. The results of these two studies indicated the potential of deep learning in extracting relevant properties from the training data even without carefully constructed features, which may render feature extraction and selection applications unnecessary in the future.

In one of the earliest applications of DNNs for DTI prediction, restricted Boltzmann machines (RBM), which is a two-layer undirected graphical model was employed [131]. An RBM is not considered as a deep architecture since it only contains one hidden layer. However, an individual RBM was generated for each target and a large network composed of multiple RBMs was implemented as the final model. The main aim in this study was to construct a multidimensional DTI network model by

incorporating DTIs from diverse set of compounds and targets with different types of interactions. The interaction types were divided between ligands and receptors into two groups as direct and indirect interactions. The physical binding of small molecule drugs to target proteins was referred to as direct interaction. The indirect interactions corresponded to the effects of the compounds on proteins by means other than direct binding (e.g., changing the expression level of the gene that encodes the target). The interaction type information was incorporated by adding edge properties to their network. Besides, additional models were constructed for predicting drug modes of action (e.g., activation and inhibition). DTI information in the MATADOR and STITCH databases were used for the training and testing of their method, and it was found that the method was able to predict different types of DTIs and drug modes of action with high accuracy. The proposed method was compared with a simple logic-based approach, and it performed better. Finally, new DTI predictions were produced using the proposed method, and verified through literature evidence.

DeepDTIs were developed for the prediction of drug-target interactions using deep belief network (DBN), which is constructed by stacking multiple Restricted Boltzmann Machines (RBM) [130]. In DeepDTIs, targets are not separated into classes according to protein families to train individual models, instead all targets in the training data are pooled to train one predictive model. The training data was composed of drug-target interactions from the DrugBank database (i.e., 6262 DTIs between 1412 approved drugs and 1520 targets). To generate input features, ECFP fingerprints were employed for compounds and sequence composition descriptors were used for target proteins and they were all merged to represent drug-target pairs (i.e., a 14564-dimensional vector for each pair). Experimental drug-target pairs from DrugBank was used to assess the performance of DeepDTIs and to compare it with other ML methods (i.e., Bernoulli naive Bayesian model, decision trees and random forests). The method was also applied to predict the unknown DTIs between all combinations of drug and targets in their training set and the most probable predictions were manually verified through literature-based evidence. Finally, in order to test the ability of DBN in abstracting the input and generating a more

informative representation of the data in each successive hidden layer, the transformed data generated at each layer was used to train a simple logistic regression classification model for the prediction of DTIs. The performance of the LR model increased with the increasing hidden layer depth, which indicated the effectiveness of the approach.

The method "AtomNet" by Wallach *et al.* [123] is one of the earliest applications of CNNs for structure-based virtual screening. The proposed method incorporated both the compound and target features for training by using the 3D structural information of ligand-receptor (i.e., compound-target) complexes. 3D grids placed over the atomic coordinates in the ligand-receptor complexes were used as input to their CNN, where each grid contained numerical structural features such as atom type enumerations and structural protein-ligand interaction fingerprints. Three datasets (i.e., the DUD-E set and two generated datasets: a DUDE-like benchmark set composed of 78,904 active compounds, 2,367,120 inactive compounds and 290 targets and another dataset with experimentally-verified inactive molecules composed of 78,904 active compounds, 363,187 inactive compounds for 290 targets, both constructed using ChEMBL) were employed to train and validate their method. For the training of the system, targets that have at least one annotated binding site in sc-PDB database were used. The prediction results were compared with two state-of-the-art structure-based virtual screening (i.e., docking) methods using abovementioned datasets and the described method outperformed the other algorithms with a large margin. This study is significant in terms of indicating that CNNs can be used to model the structural properties of ligand-receptor complexes with a performance better than conventional docking based approaches.

A CNN architecture with a mixture of PCM and structure-based DTI prediction approach was also proposed [132]. The method takes protein 3D structure information (i.e., the specific binding pocket of the target) along with compound descriptors (i.e., fixed-size 3D structural fingerprints based on learnable atom convolution operations generated from ECFPs) in a pairwise-input format. The insufficiency of current benchmarking datasets for testing structure-based methods

was discussed and instead, a new dataset generated from DUD-E, PDBBind and MUV datasets was described. The method was trained and tested by this described dataset. The method was compared with the state-of-the-art methods (i.e., docking methods and AtomNet: another DNN-based approach) and the models trained with learnt compound features resulted in better performance compared to the models trained with simple ECFPs.

Another CNN based method for the prediction of chemical properties of compounds such as binding, toxicity and free energy solvation was described by Goh *et al.* [133]. CNN-based techniques are highly utilized in computer vision with high performance. The focus of this study was constructing predictive models with minimal amount of feature engineering and chemical knowledge. In this method, each compound was represented as an 80x80 pixel sized image based on their 2D drawings as shown in chemical databases. These images were then fed to the CNN for classification. Three different datasets were obtained from MoleculeNet benchmark database. The first dataset was Tox21, which was composed of 8,014 compounds labelled as "toxic" or "non-toxic". The second dataset was freeSolv dataset, including 643 compounds with measured hydration free energies of small-molecules. Lastly, HIV dataset included bioactivity measurements of 41,913 compounds against the inhibition of HIV replication. Two classification models were separately trained using HIV and Tox21 datasets and a regression model was trained using freeSolv dataset. The results were compared with the results of the models that were trained with conventional ECFP4 fingerprints using multi-task DNNs. The descrobed method slightly outperformed the conventional feature utilization method in HIV and freeSolv datasets and slightly underperformed in Tox21 dataset.

A graph convolution deep learning method was described to extract learnable features from the graph representations of compounds (the vertices in the graphs correspond to atoms and edges correspond to bonds between atoms) and to perform learning using the extracted features for DTI prediction [134]. Several datasets coming from PubChem, Tox21, MUV and DUD-E were combined to achieve a total of 38 million data points. The graph structures of compounds were generated using

SMILES representations and the extracted graphs were fed to the proposed DNN to train the system. The described models were compared with the models trained with multi-task DNN, random forest and logistic regression methods, which were trained using ECFP4 fingerprints. The described method could not outperform the other methods but achieved a comparable performance. Nevertheless, this work stands as a proof of concept that indicates graph convolutions can be a good alternative for employing deep learning for virtual screening with a simple compound feature encoding.

A novel deep-learning architecture "iterative refinement long short-term memory" (IterRefLSTM) was developed using graph convolutional neural networks especially for protein targets with low number of training instances [135]. The method allows the learning of sophisticated small molecule features using one-shot learning methodology, and yield more reliable predictions when the training dataset is small. Training datasets were generated using assay results from three different sources, which were Tox21 challenge dataset, SIDER database and Maximum unbiased validation (MUV) dataset [120], [137], [138]. Drug-target prediction problem was designed again as a binary classification problem and multiple models were trained for each target, where inputs were 2D graph structures of compounds and outputs were binary variables as active or inactive. One-shot deep learning architecture was combined with iterative refinement long short-term memories and graph convolutions. Graph convolutional features of compounds were used as feature vectors to train neural network models. This novel method was compared with random forest as a baseline classifier. The proposed method obtained significant performance improvement on datasets having low number of training samples compared to the baseline classifier. The models were released as a part of the open-source DeepChem framework (https://github.com/deepchem/deepchem).

DeepDTA is a non-structure based binding affinity prediction method that employs pair-wise input deep neural network. In DeepDTA, an integer value is assigned for each possible symbol in SMILES notation and for each amino-acid and the input compounds and protein sequences are encoded based on the assigned values for

SMILES symbols and amino acids, respectively. The encoded ligand and protein representations are fed to embedding layers which are followed by 1-D convolutional and pooling layers. The last convolutional layers are flattened and concatenated and the concatenated layer is connected to fully-connected layers. The output of DeepDTA is the predicted binding affinity prediction value. The aim of the method is to minimize the difference between the predicted and measured binding affinity values. They used two different datasets (Davis [139] and KIBA [140] datasets) to compare their method with state-of-the-art methods and showed that their method outperformed the other methods.

DeepConvDTI is another method that uses pairwise input deep neural network. In this method, the aim is to predict binary drug-target interaction prediction based-on predefined active and inactive drug-target pairs. This method transforms raw protein sequences into learnable embedding vectors which are passed through convolutional layers. Ligands are represented as 2,048 dimensional Morgan/Circular fingerprints and the constructed feature vectors (i.e. fingerprints) are fed to fully-connected layers. The last layer of convolutional layer and the last layer of fully-connected layer are concatenated to construct a fully-connected layer. The output of the method is the binary drug-target interaction (active or inactive) for the input drug-target pair. The active and inactive data pairs for training are collected from three databases which are DrugBank [141], International Union of Basic and Clinical Pharmacology (IUPHAR) [142], and Kyoto Encyclopedia of Genes and Genomes (KEGG) [83]. Authors also created independent active and inactive test datasets from PubChem Bioassays datasets.

According to the "deep learning for virtual screening" studies published so far, DNNs are especially convenient for analysing the relationship between the compounds and targets since the data is high-dimensional and the attributes contributing to molecular interactions are not clearly known [116]. In these studies, the deep models have exhibited elevated drug-target interaction prediction performance even with minimal data pre-processing and minimal parameter optimization. In these works, the authors mostly focus on discussing the applicability

of deep learning techniques on drug-target interaction prediction problem over the architecture and hyper-parameter selections [121], [123], [124], concluding that deep learning has a substantial potential to advance the field of computational drug discovery [117], [118].

Apart from DTI prediction, deep learning techniques are also employed for other drug discovery related purposes. For instance, Mayr *et al*. developed DeepTox, an ensemble deep learning based compound toxicity prediction method and won the Tox21 data challenge [120]. Related to this, Maltarollo *et al*. reviewed the applications of various machine learning approaches including DNNs for ADME-Tox (i.e., absorption, distribution, metabolism, excretion and toxicity) prediction [143]. Aliper *et al*. proposed a DNN-based therapeutic effect predictor for compounds, using only the drug-induced transcriptomic profiles in different cell lines as input [144]. In one of the earliest applications of deep learning in drug discovery Lusci *et al*. proposed an ensemble of recursive neural networks to predict the molecular properties of compounds such as the aqueous solubility. The authors developed a web-based tool "AquaSol" for the prediction of the aqueous solubility of compounds, which takes SMILES representations as input [145].

There are several review articles on deep learning applications on the biomedical data [69], [113], [114], [118], [119], [133], [146], [147]. In some of these studies, the authors explained several DNN architectures that has been successfully applied on non-biomedical fields and discussed the current and potential applications on biomedicine [69], [113], [114], [118], [119], [148]. In a few of these review studies, specific applications of DNNs in virtual screening has been discussed as well [118], [147], [148]; however, most of the original research articles on this topic came out just recently (in late 2016 and in 2017), which were not included in these reviews. Apart from the machine learning based prediction methodologies, some review studies focused on available toolkits, frameworks, databases and representations/descriptors for computational drug discovery [39], [43], [88], [148].

2.1.4 Evaluation Metrics and Performance Comparison of Virtual Screening Methods

Evaluating the performance of machine learning methods is crucial to be able to assess how well a method performs, and to fairly compare the performances of different methods. Here, we demonstrate the most widely used evaluation metrics in the literature, which are precision, recall, F1-score, F0.5-score, accuracy and Matthews correlation coefficient (formulations are given below together with quantitative ranges).

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

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

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$F0.5\ score = \frac{1.25 \times Precision \times Recall}{0.25 \times Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

$$False\ positive\ rate\ (FPR) = \frac{FP}{FP + TN}$$

$$AUROC = Area\ under\ the\ receiver\ operating\ characteristic\ curve$$

In the equations above, TP, FP, TN and FN represent he number of true positives, false positives, true negatives and false negatives, respectively. Each of these metrics have different properties. For example, precision refers to fraction of the correctly predicted samples (TP) among all positively predicted targets, whereas recall (i.e., true positives rate) denotes the fraction of correctly predicted samples among all truly positive samples. Evaluating the performance of methods using only precision or only recall may result in unrealistic conclusions. For example, if using only precision as the evaluation metric would results in overlooking the high number of FN predictions, since precision does not take false negatives into account. The same case is applied for the recall and the false positives. To overcome this issue, F1-score is employed, which is a harmonic mean of precision and recall, to consider both the FPs and FNs. F1-score gives equal weights to precision and recall, therefore both metrics are treated same. However, in some virtual screening studies, reducing the number of FPs is considered to be an important issue to provide more reliable predictions [101], [102]. For this, F0.5-score is used, where twice the weight is given to precision compared to recall, in order to minimize number of FP predictions; in other words, to increase the probability of a positive prediction to be a TP. Accuracy measure can be defined as the fraction of correctly predicted samples among all samples in the training dataset. Evaluating the system performance based on accuracy may result in high bias, especially when the positive and the negative classes are imbalanced. Considering the virtual screening data, the number of negative samples are usually significantly higher than number of positive samples. For a failing predictive model which classifies all instances as negative (i.e., inactive

or non-interacting), the accuracy measure would result in overestimated performance. Matthews correlation coefficient (MCC) is another measure which also is a balanced performance calculation metric similar to the F1-score. It was reported that MCC can very well be used for performance evaluation when classes are imbalanced [149]. The main difference between MCC and F1-score is that F1-score does not take TNs into account, whereas MCC does. Therefore, using MCC for performance evaluation can be more convenient especially when one have a reliable negative training dataset. All of the metrics explained above are used to measure the performance of a predictive model at one point (i.e., at a selected prediction score threshold, above which the corresponding compound-target pair is predicted to be interacting/active, and below which they are estimated to be non-interacting/inactive). However, the generalization of the performance over the whole threshold spectrum is also required especially to fairly compare the performance of multiple methods. The area under the receiver operating characteristic (AUROC) curve (i.e., a 2-dimensional plot where the horizontal and the vertical axis correspond to false positives rate and the true positives rate, respectively; drawn considering the performance measures at different arbitrarily selected score thresholds) or the area under the precision vs. recall curve – AUPR (i.e., a similar plot where the precision and recall values are used as the 2 dimensions) are employed for this purpose. It is also important to note that the discriminative power of AUROC diminishes at low false positives rates; as a result, AUROC is usually considered inferior to AUPR. Considering the range of values that can be obtained using these metrics, 1 usually indicates a perfect classifier and the classifier performance decreases with the resulting measure getting closer to 0. As for MCC where the range is between -1 and 1, the measure of 0 indicates a random classifier and -1 indicates a perfect negative correlation. As a conclusion, the evaluation metrics should be selected based on nature of the problem at hand. Calculating the performance of different systems using multiple evolution metrics is generally preferred to be able to observe the system behavior from different perspectives.

2.2 Databases and Gold Standard Datasets for Virtual Screening

The aim of this s is to provide a brief overview of the open access chemical and biological data repositories as well as the available gold standard datasets that are widely used in virtual screening. Compound and bioactivity databases, together with the tools that they provide, are crucial for the development of novel virtual screening methods. The databases for compounds, bioactivities and proteins, and their statistics are given in Table 2.3.

2.2.1 Compound and Bioactivity Databases

With the improvements in the drug screening technologies and virtual screening methods, the amount of both the experimental bioassay data and computationally produced DTI data are increasing. Therefore, researchers require structured chemical and biological databases to store and publish this vast amount of data in a well-organized way. A *chemical database of bioactive molecules* (i.e., a compound database) is a resource that contains several properties of chemical substances such as 2D and 3D structures, physical and chemical attributes, molecular descriptors, side effects and clinical information; as well as, targets and activity measurements. The public release of large scale experimental bioactivity data (mostly from HTS assays) have started a new era in computational biomedical research. Research groups from all around the world have started to access and analyse the data, which boost the field of computational drug discovery (specifically virtual screening) in the last decade. In this sense, the prominent bioactivity and compound data resources can be listed as PubChem [1], ChEMBL [2], DrugBank [5], STITCH [150], BindingDB [151], BindingMoad [152], KEGG [83], SIDER [137], DCDB [153], HMDB [154] and T3DB [155]. Although the discussed databases have common properties, they also complement each other by providing different features. For example, PubChem contains the largest bioactivity data for compounds -mainly retrieved from HTS experiments- and the other databases generally import data from

PubChem. ChEMBL is also a large-scale compound and bioactivity database. However, one of the most significant differences of ChEMBL from the other large-scale sources is that the provided data is manually curated by experts from the literature in a comprehensive manner, making ChEMBL a more reliable resource, whereas the PubChem data is non-curated. ChEMBL also categorizes targets as "Single Protein", "Protein Family" and "Protein Complex" and assigns a confidence score to state the specificity of compound activity. The main advantage of using PubChem over the other resources is its unmatched high volume (i.e., in terms of the number of bioassays, bioactivities, compounds and targets). Another bioactivity database BindingDB contains only experimentally validated bioactivity values of compound-target complexes without considering other functional assay results. BindingDB directly provides validation data sets for computational drug design studies. In contrary to PubChem, ChEMBL and BindingDB, BindingMoad is a small-scale bioactivity database, which includes high-resolution 3D structures of proteins and their ligand annotations for related protein-ligand interactions. In this sense, BindingMoad is especially convenient to be employed for the structure based virtual screening approaches. As an extensive network of biological systems, KEGG is a valuable resource for understanding functional hierarchies of biological events involving molecular interactions, pathways and disease mechanisms from molecular-level information of genes and genomes extracted from large-scale datasets of genome sequencing or other high-throughput experimental techniques. DrugBank database includes information regarding the approved and experimental drugs along with their target associations; hence, it is a small-scale database. However, DrugBank covers almost all aspects of drugs as a manually curated biomedical resource with high-quality standards. The data obtained from DrugBank is often used in test sets for novel large-scale virtual screening methods. SIDER and STITCH are sister projects, where the former focuses on side effect information and the latter focuses on the compound-target interactions under biological networks point of view. Therefore, it is quite common to combine complementary features from these databases, when applicable. In addition to the abovementioned resources,

there are also useful databases such as DCDB, HMDB and T3DB, which focus on drug combinations, human metabolites and toxic substances, respectively. Considering these bioactivity databases, PubChem, ChEMBL, Binding MOAD and BindingDB represent activity data with quantitative measurements such as the IC50, EC50, Ki and potency; while DrugBank, STITCH, KEGG, DCDB, HMDB and T3DB only provide the information regarding presence of an activity/interaction between the corresponding drug-target pairs.

Table 2.2 Statistics of compounds, bioactivities and target protein databases

Compound& Bioactivity Databases	**Statistics***			**Version**
	Compounds	**Targets**	**Interactions**	
PubChem [1]	93,977,773(C) 235,653,627(S)	10,341(P)	233,799,255(I) 1,252,820(E)	03.12.2017
ChEMBL [2]	1,735,442(C)	11,538 (P)	14,675,320(I) 1,302,147(E)	v23
DrugBank [5]	9,591(D)	4,270 (P)	16,748(I)	v5.0
STITCH [150]	~500,000(C)	9,643,763(P)	~1,6billion(I)	v5.0
BindingDB [151]	635,301(C)	7,000 (P)	1,419,347(I)	03.12.2017
BindingMoad [152]	12,440(C)	7,599 (F)	25,769(I)	Rel. 2014
KEGG [83]	18,211(C) 10,484(D)	976 (P)	6,502(I)	Rel. 84.1
DCDB [153]	904(D) 1,363(DC)	805(P)	-	v2.0

Table 2.2 (continued)

T3DB [155]	3,673(T)	2,087(P)	42,471(I)	v2.0

SIDE Effect Databases	**Statistics***	**Version**
SIDER [137]	1,430 (D), 5,868 (SE), 139,756 (A)	v4.1

Metabolome Databases	**Statistics***	**Version**
HMDB [154]	114,089 (M)	v4.0

Chemical Databases	**Compounds**	**Version**
ChemSpider [3]	~62,000,000(C)	03.12.2017
ChEBI [4]	53,495(C)	Rel. 158
ZINC [156]	~100,000,000(C)	ZINC 15

2.2.2 Gold Standard Datasets for Virtual Screening

In machine learning, the term "gold standard datasets" refer to reliable sets of information created to address a particular problem, which can be used for the following purposes:

- Development (i.e., training and testing) of computational methods;
- adjustment of the parameters of computational methods;
- evaluation of the performance of trained models;
- benchmarking to compare the performances of various prediction models.

In virtual screening, gold standard datasets generally comprise of manually curated compound-target pairs and their bioactivity values. The abovementioned data repositories provide data that can be used for model training and benchmarking; however, it is not easy to understand which database to employ at which step, to obtain the required data. Therefore, dataset construction is one of the critical steps in virtual screening studies. Although these databases provide cross-references to each other to some extent, the data is mostly disconnected and it is often non-trivial to carry out data integration operations on different resources; which requires expert level knowledge. As a result, expert curated gold-standard datasets are extremely valuable for the community.

Due to the lack of adequate experimental data and publicly available data repositories, it was a significant problem to define a suitable gold standard dataset for benchmark studies until 10 years ago. The early datasets were either too small or proprietary. For example, a dataset generated in 1988 for comparative molecular field analysis (CoMFA) included only 21 varied steroid structures for the analysis of their binding affinities to human corticosteroid- and testosterone-binding globulins [157]. In 2001, Hert *et al.* generated a dataset for the comparison of different types of 2D fingerprints used in similarity-based virtual screening with a total of 11 activity class each of which was involving active compounds in a range of approximately 300-1200. However, this dataset was derived from MDL Drug Data Report database, which is licensed and not publicly available [158].

As one of the first gold standard datasets that is large enough and freely accessible, Yamanishi *et al.* created a dataset with four classes (i.e., families) of targets that are enzymes, ion channels, G-protein coupled receptors (GPCRs) and nuclear receptors [79]. The dataset by Yamanishi *et al.* involves only human proteins and was constructed using KEGG BRITE, BRENDA, SuperTarget and DrugBank databases; and generated mainly for evaluating and training of their own VS method. This dataset can be reached via: http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/. The numbers of targets in these datasets are 664, 204, 95 and 26; whereas the numbers of drug-target interactions are 2926, 1476, 635, 90, respectively for each

class. An updated version of the dataset was later created again by Yamanishi *et al.*, including the same target classes [86]; this time using the JAPIC database (http://www.japic.or.jp/). The numbers of the targets in the updated set are the same as previous dataset, and the numbers of the interactions are 1515, 776, 314 and 44, respectively for each class. Yamanishi's sets were generated to train and to test the performances of network/graph base DTI prediction methods; thus, they are among the most utilized benchmarking datasets for network/graph based approaches. However, they usually are not suitable for machine learning approaches, which require large training datasets. Yamanishi's gold-standard sets can be downloaded from http://cbio.mines-paristech.fr/~yyamanishi/pharmaco/.

Huang, Irwin and Shoichet have generated a benchmarking dataset called DUD (directory of useful decoys) for testing virtual screening methods, by curating challenging decoys that have a very low probability of interacting with the selected targets. The DUD dataset contained active compounds for the selected targets together with 50 decoys for each active compound, which have similar physicochemical properties but different topology [159]. As an updated and enhanced version of DUD with more diverse target classes such as GPCRs and ion channels (along with enzymes and nuclear receptors) DUD-E contains 22,886 ligands and their affinities against 102 targets retrieved from the ChEMBL database, together with property-matched decoys obtained from the ZINC database. The dataset is freely available at http://dude.docking.org [160].

Another benchmark dataset designed for virtual screening is Maximum Unbiased Validation (MUV), which was generated from PubChem bioactivity data by topological optimization based on a refined nearest neighbour analysis. MUV provides randomly distributed sets of active compounds -selected from potential actives (PA) and inactive compounds -selected from potential decoys (PD) that minimizes the influence of dataset bias on validation results. The workflow used for the generation of optimized MUV dataset is also freely available as a software package that can be applied on other activity datasets for optimization. The dataset

and the software package can be accessed via https://www.tu-braunschweig.de/pharmchem/forschung/baumann/muv [138].

In 2012, Merck sponsored a drug target interaction challenge over Kaggle data competition service (https://www.kaggle.com/c/MerckActivity). They provided 164,024 compounds for 15 biologically relevant targets. For each activity, they provided a list of chemicals along with their molecular descriptors and bioactivity measurement values. The participating teams tried to predict the experimentally known held-out interactions among the overall dataset. The evaluation mechanism and the performance results of the teams are available in the competition page. Following the end of the competition, the held-out evaluation sets were released, which can now be used as benchmarking datasets for different virtual screening approaches. The datasets are explained in the publication by Ma *et al*. [117] and available at https://www.kaggle.com/c/MerckActivity/data.

Another dataset called Tox21 is also commonly used in machine learning based computational drug discovery applications. This dataset has been generated by The Tox21 Data Challenge community in 2014 to evaluate the performances of different computational methods in terms of toxicity prediction. The dataset comprises approximately 12,000 environmental chemicals and approved drugs screened in 12 different bioassays related to nuclear receptor signaling and stress response pathways to reveal their toxic effects based on the disruption of these processes [161].

There are also novel approaches for generating gold standard datasets especially for deep learning applications in DTI prediction. Wu *et al*. developed a platform, MoleculeNet, as a benchmark collection for machine learning methods used in molecular systems. The curated dataset of MoleculeNet contains nearly 700,000 compounds retrieved from publicly available databases such as QM7/QM7b, QM8, QM9, ESOL, FreeSolv, Lipophilicity and PDBBind for regression datasets and PCBA, MUV, HIV, BACE, BBBP, Tox21, ToxCast, ClinTox and SIDER for classification datasets. The data was split into training/validation/test subsets and tested on a range of categories, such as quantum mechanics, physical chemistry,

biophysics and physiology. Furthermore, MoleculeNet provided evaluation metrics and open-source implementations of several well-known molecular featurization methods and machine learning algorithms. All parts of MoleculeNet have also been integrated into DeepChem open-source framework (https://github.com/deepchem/deepchem) [162]. Apart from these gold-standard sets, there has also been efforts to generate purpose specific data sets [163], often using the ZINC database [156] as their resource. With the increased volume of open access experimental data in repositories such as PubChem, ChEMBL, ZINC and etc., the data resources for virtual screening studies has been significantly changed, compared to 10 years ago. Novel datasets derived from these resources such as the DUD and MUV, together with the new algorithmic approaches, are highly promising in terms of developing the field of computational drug discovery. The field of generating and utilizing gold-standard/benchmarking datasets for virtual screening has been extensively discussed in the recent works by Lagarde *et al*. and Xia *et al*. [163], [164].

CHAPTER 3

DEEPred: AUTOMATED PROTEIN FUNCTION PREDICTION WITH MULTI-TASK FEED-FORWARD DEEP NEURAL NETWORKS

3.1 Chapter Overview

[3]Automated protein function prediction is critical for the annotation of uncharacterized protein sequences, where accurate prediction methods are still required. Recently, deep learning based methods have outperformed conventional algorithms in computer vision and natural language processing due to the prevention of overfitting and efficient training. Here, we propose DEEPred, a hierarchical stack of multi-task feed-forward deep neural networks, as a solution to Gene Ontology (GO) based protein function prediction. DEEPred was optimized through rigorous hyper-parameter tests, and benchmarked using three types of protein descriptors, training datasets with varying sizes and GO terms form different levels. Furthermore, in order to explore how training with larger but potentially noisy data would change the performance, electronically made GO annotations were also included in the training process. The overall predictive performance of DEEPred was assessed using CAFA2 and CAFA3 challenge datasets, in comparison with the state-of-the-art protein function prediction methods. Finally, we evaluated selected novel annotations produced by DEEPred with a literature-based case study considering the 'biofilm formation process' in *Pseudomonas aeruginosa*. We participated CAFA PI challenge and ranked fourth in *Motility* prediction category of the challenge. This study reports that deep learning algorithms have significant potential in protein function prediction; particularly when the source data is large. The neural network

[3] The main content of this chapter was published in *Scientific Reports* journal in 2019 [191]. Please note that only parts that I worked on were included from the corresponding publication.

architecture of DEEPred can also be applied to the prediction of the other types of ontological associations. The source code and all datasets used in this study are available at: https://github.com/cansyl/DEEPred.

This chapter consists of the parts that I mainly worked in DEEPred. The rest of the conducted research and analysis can be reached from our publication. My specific contributions in DEEPred are listed below:

- Development and design of the proposed method;
- Implementation of the overall system;
- Investigation of state-of-the-art methods and benchmarking datasets;
- Training of models for CAFA PI challenge and running models on target datasets;
- Implementation of scripts for the analysis and discussions.

3.2 Introduction

Functional annotation of proteins is crucial for understanding the cellular mechanisms, identifying disease-causing functional changes in genes/proteins, and for discovering novel tools for disease prevention, diagnosis, and treatment. Traditionally, gene/protein functions are first identified by *in vitro* and *in vivo* experiments and recorded in biological databases via literature-based curation. However, wet-lab experiments and manual curation efforts are cumbersome and time consuming. Thus, they are unable to resolve the knowledge gap that is being produced due to the continuous growth of biological sequence data [165]. Therefore, accurate computational methods have been sought to automatically annotate functions of proteins.

The Gene Ontology (GO) provides a controlled vocabulary to classify the attributes of proteins based upon representative terms, referred to as "GO terms" [166]. The GO system divides protein attributes into three main categories: molecular function (MF), biological process (BP) and cellular component (CC). Each GO term

represents a unique functional attribute and all terms are associated to each other in a directed acyclic graph (DAG) structure based on inheritance relationships. Several GO term-based protein function prediction methods have been proposed in the last decade to automatically annotate protein sequences using machine learning and statistical analysis techniques [167]–[172]. Considering the prediction performances of the current methods, it can be stated that there is still room for significant improvement in this area. Critical Assessment of Protein Function Annotation (CAFA) is an initiative, whose aim is the large-scale evaluation of protein function prediction methods, and the results of the first two CAFA challenges showed that protein function prediction is still a challenging area [173], [174].

Numerous types of machine learning techniques have been employed for protein function prediction; one of which is the artificial neural networks (ANNs). ANNs can be considered as a framework of interconnected processing units, that was inspired from the central nervous systems of animals and are usually employed to process complex data inputs[175]. ANNs can be constructed as classifiers to distribute query instances to pre-defined classes. In this case, the ANN accepts a feature vector as input and applies nonlinear transformations before providing a class prediction as output. ANNs consist of a single input layer and a single output layer along with one or more intermediate layers called ‘hidden layers’. Each layer includes a certain number of nodes (i.e., neurons or processing units), which are linked to the nodes of the next layer via a system of weighted connections, to transmit signals. Deep Neural Network (DNN) algorithms, a sub-group of ANNs, have multiple hidden layers. DNNs take low level features as input and build more advanced features at each subsequent layer. DNN-based methods have already become industry standards in the fields of computer vision and natural language processing [112], [114], [176]–[178]. Recent improvements in affordable computational power have allowed the scientific community to apply DNN-based methods on numerous research fields including biomedical data analysis; where, DNN algorithms have been shown to outperform the traditional predictive methods in bioinformatics and cheminformatics [115], [118]–[120], [179]. DNNs are divided

into two groups in terms of the task modelling approach. Multi-task DNNs are designed for classifying the input instances into multiple pre-defined classes/tasks [180], as opposed to single-task DNNs, where the aim is to make a binary prediction. In terms of the model architecture and properties, DNNs are classified into multiple groups, the most popular architectures are feed-forward DNN, recurrent neural network (RNN), restricted Boltzmann machine (RBM) and deep belief network (DBN), auto encoder deep neural networks, convolutional neural network (CNN), and graph convolutional network (GCN) [114], [118], [119], [177], [180], [181].

Investigative studies showed that, applications of multi-task DNNs provided a significant performance increase in ligand-based drug discovery. Ligand-based drug discovery can be considered similar to the problem of protein function prediction [63], [179]; where in protein function prediction, the identification of associations is between the ontology-based function defining terms (e.g., GO terms) and a protein, and a protein may have more than one function. Therefore, protein function prediction is a multi-label learning problem and thus can be solved using multi-task deep neural networks similar to the drug discovery [54]. Multi-task DNN algorithms inherently extract the relationships between multiple classes by building complex features from the raw input data at each layer in a hierarchical manner. Additionally, shared hidden units among different classes enhance the prediction results of the classes that have a low number of training samples, which often has a positive impact on the predictive performance.

To the best of our knowledge, deep learning algorithms have not been thoroughly investigated in terms of generating practical large-scale protein function prediction pipelines. However, there have been a small number of studies mostly confined to small sets of proteins and functional classes. In these studies, DNNs were applied to protein function prediction using different types of protein features such as amino acid sequences [182]–[185], 3-D structural properties [186], protein-protein interaction networks [184], [187] or other molecular and functional aspects [185], [188]–[190], and various types of DNN architectures such as single or multi-task feed-forward DNNs [188], recurrent neural networks [182], [183], deep autoencoder

neural networks [187], [189], deep restricted Boltzmann machines [190] or convolutional neural networks [184]–[186].

One of the most critical obstacles against developing a practical DNN-based predictive tool is the computationally intensive training processes that limits the size of input data and the number of functional categories that can be included in the system. Due to this reason, previous studies mostly focused on a small number of protein families or GO terms. Whilst methods covering large sets of GO terms suffered from long training duration and reduced predictive performance issues. Therefore, there is a need for new predictive methods not only with high performance, but also with real-world usability, to be able to support *in vitro* studies in protein function identification.

In this study, we propose a novel multi-task hierarchical deep learning method, DEEPred, for the prediction of GO term associations to protein sequence records in biological data resources such as the UniProtKB, as well as for poorly and uncharacterized open reading frames. We also provide a comprehensive investigation on DNN-based predictive model characteristics when applied on protein sequence and ontology data.

3.3 Materials and Methods

3.3.1 Training Dataset Construction

The training dataset was created using the UniProtKB/Swiss-Prot database version 2017_08 protein entries. UniProt supports each functional annotation with one of the 21 different evidence codes, which indicate the source of the particular annotation. In this study, we used annotations with manual curation or experimental evidences, which are considered to be highly reliable. In order to generate the training dataset, the corresponding annotations were extracted from the UniProt-GOA database, propagated to their parent terms according to the "*true path rule*", which defines the

inheritance relationship between GO terms [166]. Using this dataset, a positive training dataset was constructed for each GO term. In short, proteins that are annotated either with the corresponding GO term or with one of its children terms, were included in the positive training dataset of the corresponding GO term. Since our multi-task DNN models are composed of multiple GO terms, the positive training dataset for one GO term, in a model, constitute the negative training dataset of the other GO terms in the same model, except the proteins that are annotated with both GO terms.

In order to analyze the effect of the extent of training datasets on the predictive performance, we constructed multiple "training-set-size-based" datasets, taking into account the number of protein associations of GO terms. For example, one of our training-set-size-based datasets includes all GO terms that have more than or equal to 30 protein associations. Hence, we created six different datasets, where GO terms in each dataset have greater than 30, 100, 200, 300, 400 and finally 500 protein associations respectively. These datasets comprise each other (e.g., GO-terms-with-greater-than-30-proteins dataset covers the GO-terms-with-greater-than-100-proteins dataset). The statistics (i.e., number of annotations, GO terms and proteins) about these datasets are given in the Results chapter.

3.3.2 DEEPred Architecture

DEEPred was built as a stack of multi-task feed-forward deep neural networks, connected to each other. In DEEPred, each DNN was modelled to predict 4 or 5 GO terms, thus multiple DNNs were required to cover thousands of terms. Figure 3.1 displays a representative DNN model in DEEPred.

The selection of GO terms for each DNN model was based on the levels of the terms on the GO DAG. The main objective of this approach was to create a multi-task deep neural network model for each level. For this, the levels of all GO terms were

extracted and the terms were separated into groups based on the level information (via topological sorting).

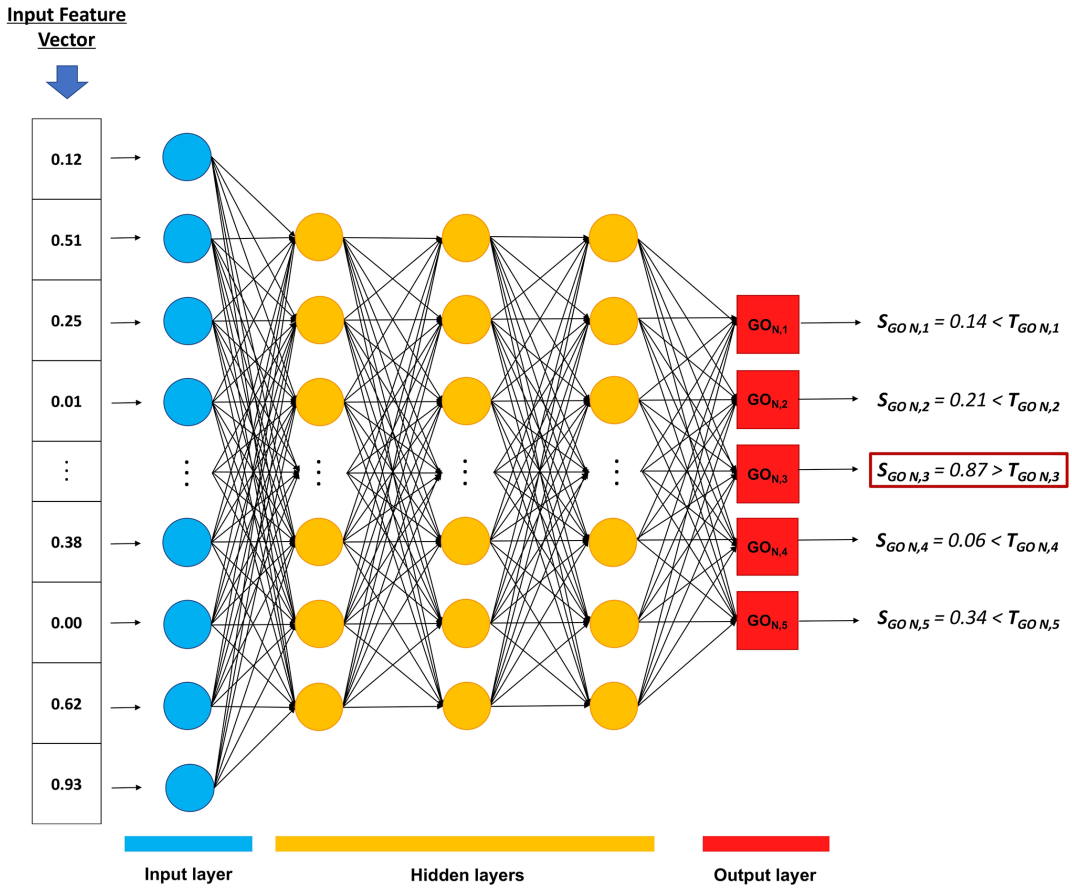


Figure 3.1. The representation of an individual multi-task feed-forward DNN model of DEEPred. Here, each task at the output layer (i.e., red squares) corresponds to a different GO term. In the example above, a query input vector is fed to the trained model N and a score greater than the pre-defined threshold is produced for GON,3, which is marked as a prediction. (Adapted from our publication [191])

We started the level numbering from generic terms; thus, they received low numbers (e.g., level 1, 2, 3, …) and the levels of specific terms received high numbers (e.g., level 10, 11, 12, …). In most cases, the number of protein associations of GO terms within a level were highly variable; therefore, we created subgroups to further avoid bias (i.e., tendency of a classifier to give predictions to classes with significantly

higher number of training instances). Here, each subgroup included GO terms with similar number of annotations. Another reason behind generating multiple models under a specific GO level was the high number of GO terms. According to our tests, when the number of tasks under a model exceed 5 or 6, the models usually perform poorly. Due to this reason, we limited the number of tasks under a model to 5 in most cases. This procedure generated 1,101 different models concerning all GO categories. Figure 3.2 represents the GO-level-based arrangement of the individual DNN models in DEEPred.

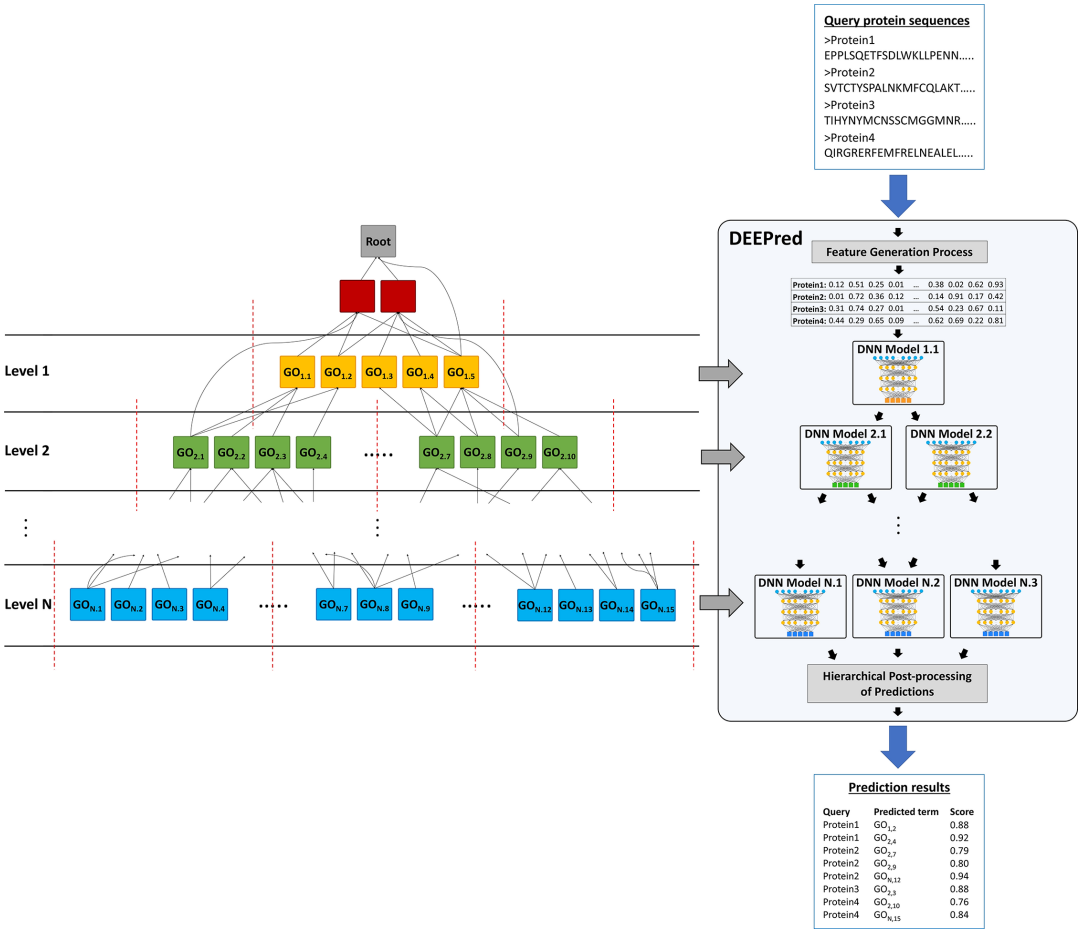


Figure 3.2. Illustration of the GO-level-based architecture of DEEPred on a simplified hypothetical GO DAG. We omitted highly generic GO terms (shown with red colored boxes) at the top of the GO hierarchy (e.g., GO:0005488 - Binding) from our models, since they are less informative and their training datasets are highly

heterogeneous. In the illustration, DNN model 1.1 incorporates GO terms: GO1,1 to GO1,5 from GO-level 1. In the real application, most of the GO levels were too crowded to be modeled in one DNN; in these cases, multiple DNN models were created for the same GO level (red dashed lines represent how GO terms are grouped to be modeled together). In this example, DNN models N.1, N.2 and N.3 incorporates GO terms: GON,1 to GON,5, GON,6 to GON,10, GON,11 to GON,15; respectively, due to the high number of GO terms on level N. At the prediction step, when a list of query sequences is run on DEEPred, all sequences are transformed into feature vectors and fed to the multi-task DNN models. Afterwards, GO term predictions from each model are evaluated together in the hierarchical post-processing procedure to present the finalized prediction list. (Adapted from our publication [191])

In a feed-forward DNN, forward propagation (z) for the layer l is calculated by the following equation:

$$z^l = b^l + W^l * a^{l-1}$$

where b^l is the bias vector, W^l is the weight matrix for the l^{th} layer and a^{l-1} is the activation value vector of the neurons at the previous layer. Subsequently, an activation function, $g^l(*)$, is applied to the calculated z^l vector and the result of the activation function is used to compute the outputs of the l^{th} layer:

$$a^l = g^l(z^l)$$

$$g^l = max(0, z^l)$$

There are alternative activation functions such as sigmoid, *tanh* and rectified linear unit (ReLU). Here, we employed ReLU activation function for the hidden layers. The prediction scores are calculated by applying *softmax* function to the neurons at the output layer. The score for the j^{th} task is calculated by the following equation:

$$S_j = \frac{e^j}{\sum_k e^k}$$

where *k* is the number of tasks to be trained within a model. At the end of the forward propagation step, prediction scores are used to calculate a cost function, *C*, based on the labels of the input samples. In this study, we used cross entropy to calculate the cost function. Once the cost function is calculated, it is used to determine how much the weights (*w*) will be changed after the last iteration by taking the partial derivatives of the cost function with respect to the weights:

$$w_i = w_i - \eta \frac{\partial C}{\partial w_i}$$

where η is the learning rate in the equation. The forward and back propagation steps were performed until the stopping criteria was met (e.g., after certain number iterations or after the objective performance is reached). For the training of the models of DEEPred, 100 iterations were selected.

In DEEPred, each model was independently trained using the feature vectors of the proteins annotated with the corresponding GO terms of that model. Considering the technical work to accomplish the multi-task training, we created a binary "true label vector" for each protein sequence using one-hot encoding, where each dimension represented a GO term to be trained in the corresponding model. The index of the GO term that was associated with the corresponding protein sequence was set to 1 and the remaining dimensions were set to 0. These true label vectors were employed to calculate the prediction errors at the output layer, which was then used by the optimizer to update weights with the aim of minimizing prediction error at each iteration.

At the prediction stage, a query protein sequence feature vector is first fed to the level 1 predictor model to receive its probabilistic scores for the corresponding GO terms and then fed to the level 2 predictor model to receive probabilistic scores for a different set of GO terms. At the end of the process, GO terms that obtained scores above the pre-determined thresholds were fed to the hierarchical post-processing (explained below under the section entitled: “Hierarchical Post-processing of Predictions”) and the finalized predictions were produced.

3.3.3 Hyper-parameters and The Optimization of Networks

A hyper-parameter is a parameter, the value of which cannot be adjusted during the training step, and thus should be selected beforehand. For this reason, the same machine learning models are trained multiple times using different hyper-parameter values, to select the ones that provided the best predictive performance. The number of hyper-parameters can be huge in deep learning algorithms; therefore, selection of best hyper-parameters is a challenge. It is known that the success of any hyper-parameter selection is dependent on the data and the architecture of the system [177]. Hence, optimal hyper-parameters should be searched for each model to be trained, individually. Below we discussed different types of hyper-parameters that we tested for DEEPred.

The most basic hyper-parameters are the number of hidden layers and the number of neurons at each of these layers. Generally, the system performance increases with the increasing number of hidden layers and the number of neurons in these layers, until it saturates at some point. However, the main disadvantage behind using excessive number of layers and neurons is the computational burden. There is a tradeoff between computational complexity, which can easily render these models impractical to run, and the predictive performance. Finding the optimal point is a challenging task and an active area of research. The strategy currently followed in the literature is testing a large number of parameter values, which was also assumed in this study.

Deep learning algorithms generally suffer from the problem of overfitting, where predictive models may perform well on training data but not on test data. Several approaches were proposed to avoid overfitting during the training of deep neural networks, known as the regularization techniques [192]–[194]. One of the most popular regularization techniques is the dropout method [193]. Dropout method randomly removes some of the neurons from different layers along with their connections at every iteration during the training procedure, so that the system is directed to find a more generalized state, that is not dependent on a few neurons and

connections. Another widely used regularization technique is the input normalization, where input features are normalized to zero mean and a variance of one. Batch normalization is another method that was proposed to reduce the effect of parameter initialization, to speed up the training and to reduce overfitting. Batch normalization is similar to input normalization; however, the aim here is the normalization of inputs of each hidden layer instead of the normalization of the input feature vectors [195].

Optimization algorithms are used to minimize an objective function, which includes learnable parameters (i.e., weights and biases) of a deep learning system. The learnable parameters are updated at each iteration using the optimization algorithm so that the system converges to an optimal solution. Several optimization algorithms were proposed in recent years, each containing one or more hyper-parameters [177]. Most widely used optimization algorithms are ADAM, RMSProp and Momentum [196], [197]. A critical hyper-parameter that is related to the optimizer is the learning rate. Briefly, learning rate value decides how much the weights should be changed (in the direction of the gradient) at each iteration. If it is selected to be very low, the training would be more reliable; however, the training process takes longer time. If it is selected to be very high, the training would be fast but unreliable (i.e., produces low performance models). Finding the optimal point for the learning rate is critical. There are also other optimizer dependent hyper-parameters such as momentum (Momentum), beta1-beta2 (Adam) and decay (RMSProp). In this study, we used default hyper-parameters for these optimizers.

In DEEPred, the total number of possible model-training-runs was huge due to the high number of selected hyper-parameter value combinations. In order to select the hyper-parameter values, we trained three sets of GO terms, chosen from different levels of GO hierarchy, with varying number of protein associations. The aim of selecting the GO terms in this way was to come up with a small GO term set, which can represent the whole system, since employing all GO terms in this test was not possible due to extremely high computational complexity. Based on the predictive performance results of different runs, we reduced the number of hyper-parameters

into a smaller set of options for the training of the whole system. We used TensorFlow framework for training the models and all computations were distributed on 2500 CPU cores in our supercomputing cluster [198]. The statistics and results of the hyper-parameter optimization tests are explained in the Results section.

3.3.4 Protein Feature Types and Vector Generation

In order to select the best protein feature representation for DEEPred, we implemented three alternative protein descriptor generation methods: *(i)* Conjoint triad [199], *(ii)* Pseudo amino acid composition [200] and *(iii)* Subsequence profile map (SPMap) [201]. Each of these feature types were used individually to train and to test the system. The details about this analysis is given in the Results chapter. The employed protein features are explained below:

Conjoint triad feature [199] considers the frequencies of amino acid triplets (i.e., consecutive three residues on the sequence). Here, query protein sequences are encoded by the frequency of the occurrence of each triplet combination. Since the total number of combinations are quite high (i.e., 20x20x20 = 8,000), Conjoint triad considers reduced alphabet by using amino acid groups generated by considering their physicochemical properties. This way, each protein is represented as a 343-dimensional (i.e., 7x7x7) feature vector. There exist several studies in the literature that employ the conjoint triad feature [202]–[205].

Pseudo-amino acid composition (PAAC) feature [200] incorporates single amino acid frequency information (i.e., conventional amino acid composition) together with sequence correlation factors without losing the sequence-order information in a protein sequence. This method computes a set of coupling factors using the physicochemical properties of amino acids (i.e., hydrophobicity, hydrophilicity value and the side chain mass) and records them in a 50-dimensional descriptor

vector in an ordered fashion. PAAC feature has frequently been employed in the literature [206]–[210].

Subsequence profile map (SPMap) is a method for functional classification of protein sequences, based on the extraction and clustering of short sub-sequence features [201]. Here we only incorporated the sub-sequence based feature vector generation module of the SPMap method. For this, all fixed-length subsequences are extracted from the protein sequences and the extracted subsequences are grouped using a hierarchical clustering approach, based on BLOSUM-62 matrix. Finally, obtained clusters are transformed into probabilistic profiles and protein sequences are converted into feature vectors based on the distribution of their sub-sequences over the generated probabilistic profiles. The original SPMap method constructs a profile for each GO term (i.e., for each model) individually. This results in protein feature vectors with varying sizes. In this study, we modified the SPMap algorithm to generate a single reference probabilistic profile using all protein sequences in the training dataset associated with all GO terms in a specific GO category. Therefore, each protein sequence was represented by a fixed-dimensional feature vector for all models, resulting in 1893, 1861 and 1901-dimensional vectors for MF, BP and CC categories, respectively. Conjoint triad and pseudo-amino acid composition features were extracted from protein sequences using the ProtR software [211]. SPMap features were calculated using our in-house software. For all methods, we used default parameters to generate the feature vectors.

3.3.5 Determining the Probabilistic Score Thresholds

When a query protein is fed to a prediction model of DEEPred, an individual probabilistic score is calculated for each GO term (i.e., task) within that model, representing the probability of the query protein possessing the function defined by the corresponding GO term. In some cases, this can be confusing because scores are on a continuous scale (i.e., it is not clear at which point one can conclude that the query protein contains the corresponding function). Usually, the requirement from a

model is to make a binary prediction instead of producing a probability. Setting a probabilistic score threshold for each GO term at each model solves this problem. At the prediction step, if the received score is equal to or greater than the pre-defined threshold, the model outputs a positive prediction for the corresponding GO term. To determine these thresholds in a validation setting (using the hold-out validation datasets), we calculated F1-score performance values for arbitrary threshold selections using the success of the binary predictions obtained when we fed the system with protein sequences with already known labels (i.e., GO term associations). We considered each GO term separately within a model and determined an individual threshold for each term by choosing the value providing the highest F1-score. These threshold values are stored in ready-to-use predictive models of DEEPred.

3.3.6 Hierarchical Post-processing of Predictions

We implemented a methodology to eliminate the unreliable predictions by considering the prediction scores received for the parents of the predicted GO term. This way, we aimed to reduce the potential of false positive hits. The reason behind applying such a post-processing step was that, multi-task DNNs tended to classify query instances to at least one of the tasks at the output layer. Such a classification scheme would not be a problem if we could generate one model that contain all of the GO terms at its output layer. However, having thousands of nodes in the output layer would be highly impractical and thus we divided GO terms into different models. This time, the problem occurs when a query protein is fed to a model, where the protein does not contain any of the functions defined by the GO terms in the corresponding model. The model often predicts one of the unrelated GO terms for the query protein, producing a false positive. We observed that separating a false positive hit (produced this way) from a reliable prediction would be possible by checking the prediction results for the parents of the predicted GO term. If the query protein consistently received high prediction scores for most of the parent terms as

well, we can conclude that this case is probably a reliable prediction; otherwise, it may be a false positive hit.

To construct this methodology, we first topologically sorted the DAG for each GO category and determined all possible paths from each GO term to the root of the corresponding category, and stored this information. When a query protein is run on DEEPred, its feature vector is fed to all trained models to obtain the prediction scores for all GO terms. Starting from the most specific level of GO, the method checks whether the prediction score of the query protein is greater than the previously calculated score thresholds. If the prediction score of a target GO term is greater than its threshold, the method checks the scores it received for the parent terms on all paths to the root, using the previously stored possible-paths-to-root. If the prediction scores given to the majority of parent terms are greater than their individual thresholds, the method presents the case as a positive prediction. This procedure is represented in Figure 3.3 with a toy example.

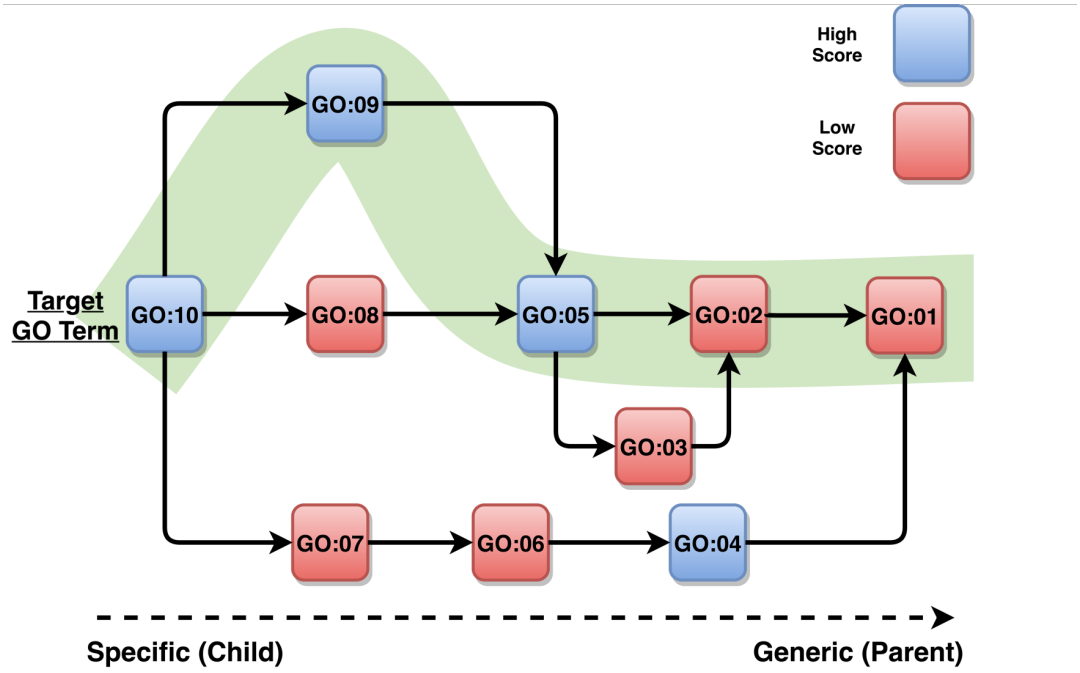


Figure 3.3. Post-processing of a prediction (GO:10) for a query protein sequence on a hypothetical GO DAG. Each box corresponds to a different GO term, with identification numbers written inside. The blue colored boxes represent GO terms

whose prediction scores are over the pre-calculated threshold values (i.e., predicted terms), whereas the red colored boxes represent GO terms, whose prediction scores are below the pre-calculated threshold values (i.e., non-predicted terms). The arrows indicate the term relationships. There are four different paths from the target term (i.e., GO:10) to the root (i.e., GO:01) in this hypothetical DAG. Since there is at least one path, where the majority of the terms received higher-than-threshold scores (shown by the shaded green line), the target term GO:10 is given as a finalized positive prediction for the query sequence. (Adapted from our publication [191])

3.3.7 Predictive Performance Evaluation Tests

According to the current deep learning practice, it is not feasible to carry out a fold-based cross validation analysis, since it usually requires extremely high computational power. This issue was also valid for DEEPred due to the presence of elevated number of models. For this reason, the assessment of DEEPred system was performed using two datasets: *(i)* Hold-out validation dataset, and *(ii)* CAFA2 challenge benchmark dataset.

The hold-out validation aims to determine and fine tune the hyper-parameter values and to observe the performance of the system. The hold-out datasets were constructed as follows: the training dataset for each GO term (see the "Training Dataset Construction" section) was randomly divided into two datasets such that 80% of the annotations were reserved for the training and 20% of the samples were used for the hold-out validation dataset. The proteins in the validation dataset were fed to the trained models to produce GO term predictions. We then compared the resulting predictions with the true annotations of these proteins to calculate the performance metrics.

We used CAFA2 challenge benchmark dataset for the independent performance evaluation and for the comparison with the state of the art methods (i.e., the methods participated to the CAFA2 challenge). Since there is a temporal difference between

CAFA2 challenge and the date we trained our system, (to yield a fair comparison) we had to remove the training instances (i.e., annotations) that were released in UniProt-GOA after the CAFA2 participation deadline, from our training dataset, and re-train our system. We then directly fed CAFA2 challenge benchmark dataset proteins as query instances to our trained models (the annotations in the CAFA2 benchmark dataset did not overlap with our training datasets). The CAFA2 benchmark dataset contained 1,828 proteins, 997 GO terms and 3,187 annotations for MF; 2,618 proteins, 3,375 GO terms and 7,956 annotations for BP; 2,938 proteins, 587 GO terms and 5,085 annotations for CC category of GO.

3.3.8 Performance Evaluation Metrics

Recall, precision, F-max and Smin measures, which are given below; were used to evaluate the performance of the system. *TP, FP, TN* and *FN* represent the number of true positives, false positives, true negatives and false negatives; respectively.

$$Precision_{\tau i} = \frac{TP_{\tau i}}{TP_{\tau i} + FP_{\tau i}}$$

$$Recall_{\tau i} = \frac{TP_{\tau i}}{TP_{\tau i} + FN_{\tau i}}$$

$$F_{max} = \max_{i=1...N} \left\{ \frac{2 * Pr_{\tau i} * Rc_{\tau i}}{Pr_{\tau i} + Rc_{\tau i}} \right\}$$

$$S_{min} = \min_{i=1...N} \left\{ \sqrt{{Ru_{\tau i}}^2 + {Mi_{\tau i}}^2} \right\}$$

In equations; τ_i represent the i^{th} probabilistic score threshold. Fmax correspond to the maximum of the F1-score values, calculated for each arbitrarily selected probabilistic score threshold. i=1...N represents there are N different arbitrarily selected probabilistic score thresholds. $Ru_{\tau i}$ and $Mi_{\tau i}$ corresponds to remaining uncertainty and normalized misinformation, respectively. Smin is the minimum semantic distance.

For CAFA2 and CAFA3 benchmark tests, we calculated the F-max scores in exactly the same way as it was described in CAFA2 GitHub repository [212]. Performance

evaluation scripts released by CAFA team were directly used for this purpose. Additional information regarding CAFA performance measures and scripts can be obtained from Jiang *et al.* [173].

3.4 Results

3.4.1 Input Feature Type Performance Analysis

Input data was quantized as feature vectors in the predictive models. These feature vectors are required to reflect the intrinsic properties of the samples they represent, which should also be correlated with their known labels (i.e., GO function terms in our case). For this reason, finding and generating the best representative feature type is important for any machine learning application. In this analysis, our aim was to investigate the best representative feature type, to be incorporated in DEEPred. For this purpose, we randomly selected three DEEPred DNN models that contain MF GO terms from different levels on the GO hierarchy, and trained each model using three different feature types (i.e., SPMap, pseudo amino acid composition - PAAC and conjoint triad) as explained in the Methods. The reason behind using MF GO term models was because molecular function is the most clearly defined aspect of GO and also the easiest one to predict.

We measured the performance of the models using cross-validation settings at 80% to 20% separation of the source training data to observe the best representative feature. Table 3.1 shows the selected models together with the incorporated GO terms, their GO levels, the number of annotated proteins and the performances for each feature type. The performances (F1-score) were calculated as 0.63, 0.36 and 0.43 for SPMap, PAAC and conjoint triad features, respectively. Since the predictive performance with SPMap feature was the best, we incorporated SPMap into the DEEPred system for the rest of the study.

Table 3.1 Input feature type performance comparison results.

Model & GO level	GO term id	# of annotated proteins	Predictive performance (F1-score)		
			SPMap	Pseudo-amino acid composition	Conjoint triad
Model 1 (GO level: 2)	GO:0036094	1 847	0.49	0.29	0.23
	GO:0003700	1 652			
	GO:0004872	1 332			
	GO:0044877	1 296			
	GO:0097367	1 252			
Model 2 (GO level: 4)	GO:0004529	50	0.68	0.53	0.38
	GO:0045309	50			
	GO:0008395	49			
	GO:0008649	49			
	GO:0015645	49			
Model 3 (GO level: 7)	GO:0001012	818	0.74	0.53	0.47
	GO:0016887	764			
	GO:0046873	685			
	GO:0001159	504			
	GO:0015077	480			

For all models, large-scale hyper-parameter optimization test served as a preliminary elimination analysis to reduce the training run times. As observed from Table 3.2, the average performances were close to each other in most cases. This was mainly due to selecting hyper-parameter values that are frequently employed in the DNN literature. The reason behind selecting 2 hidden layers instead of 3 was that, the observed performance gain was not sufficient to compensate for the increased computational run times. The learning rate 0.01 was selected among 0.001 and 0.0005 even though it produced an inferior average performance; however, in some of the models 0.01 produced significantly better results compared to the other values. Considering the number of neurons at each hidden layer, values such as (600,400), (2200,600), (200,25) etc. were selected as pairs for the first and second hidden layers, respectively.

3.4.2 DEEPred Hyper-parameter Optimization Results

The average performance result for each hyper-parameter selection is shown in Table 3.2. For example, the performance value given for the dropout rate 0.6 is the average performance of all the tests, where the dropout rate was kept constant at 0.6 and the rest of the parameters changed across the given ranges in Table 3.2. This way, different values of the same hyper-parameter became comparable to each other. Selected hyper-parameter values at the end of this performance test are highlighted with bold font. For some of the hyper-parameters, more than one value has been selected. This means that, all of these selected values were used during the training of the whole system, and the value that provide best training performance was finally selected for the corresponding model.

Table 3.2. Hyper-parameter names, their ranges used in this study and the optimization test performance results.

Hyper-parameter Name	**Range**	**Average Model Perf. (F1-score)**
Input Normalization	**Yes**	0.50
	No	0.50
Learning rate	**0.0005**	0.53
	0.001	0.52
	0.01	0.47
	0.1	0.43
Number of hidden layers	**2**	0.50
	3	0.51
# of neurons at each layer	100, 200, 400, 1000, , …	Different combinations

Table 3.2. (continued)

Optimizer	**Adam (default)**	0.51
	Momentum (default)	0.48
	RMSprop (default)	0.52
Mini-batch size	**32**	0.51
	64	0.50
Drop-out rate	0.6	0.49
	0.8	0.51
Batch Normalization	**Yes**	0.53
	No	0.47

3.4.3 Effect of Training Dataset Sizes on System Performance

DNN models usually require a high number of training instances in order to produce accurate predictions. Large-scale biological training datasets are not generally available in most cases. One solution to this problem would be to discard GO terms with a low number of training instances from the system. In this case, the problem is that there are only a small number of GO terms available for prediction, most of which are shallow (i.e., generic terms). In order to investigate the effect of training dataset sizes on the predictive performance, we carried out a detailed analysis with multiple training and testing processes.

Table 3.3 Statistics for the training datasets created by only using annotations with manual experimental evidence codes and the training datasets created by using annotations with all evidence codes.

		Manual experimental evidence codes			Annotations with all evidence codes		
	Annot Count	# of levels	# of GO terms	# of annotations	# of levels	# of GO terms	# of annotations
MF	**≥ 30**	9	838	281 125	11	2 776	6 451 530
	≥ 100	9	605	272 235	10	1 598	6 386 105
	≥ 200	9	395	257 404	10	1 174	6 326 109
	≥ 300	8	226	233 476	9	942	6 269 643
	≥ 400	8	165	218 591	9	809	6 223 762
	≥ 500	8	142	210 790	9	698	6 173 867
BP	**≥ 30**	10	4 215	1 433 220	12	8 404	16 537 812
	≥ 100	10	2 993	1 386 588	12	4 768	16 335 538
	≥ 200	9	1 782	1 302 577	11	3 299	16 129 271
	≥ 300	9	1 059	1 199 604	10	2 631	15 965 583
	≥ 400	8	743	1 123 037	9	2 233	15 828 012
	≥ 500	8	603	1 075 353	9	1 978	15 713 431
CC	**≥ 30**	7	606	340 995	8	1 268	4 167 000
	≥ 100	6	460	335 445	8	750	4 138 327
	≥ 200	6	324	325 687	7	549	4 110 383
	≥ 300	6	206	309 390	6	442	4 083 834
	≥ 400	6	155	296 929	6	377	4 061 654
	≥ 500	5	118	283 616	6	335	4 043 150

We constructed 6 different training datasets based on the annotated protein counts of different GO terms, as described in the Methods. Table 3.3 summarizes the training datasets' sizes and contents based upon molecular function annotations. There are two vertical blocks in Table 3.3, the first one belongs to "Annotations with only manual experimental evidence codes"; and the second block belongs to "Annotations with all evidence codes". As observed from the first block, the number of available GO levels and GO terms decreases as the minimum compulsory number of annotations increases, since specific GO terms usually have less number of annotations. We trained the DEEPred system with each of these training datasets (i.e., annotations with only manual experimental evidence codes) and measured the predictive performance individually. And, then compared them with each other to observe if there is a correlation. The average performance of the models for each training dataset is given in Table 3.4 and Figure 3.4. Each column in Table 3.4 corresponds to an average F1-score value of the GO terms belonging to a particular training dataset. Box plots in Figure 3.4 additionally displays median and variance values. There is a strong correlation between the training sample size and performance. As expected, increasing the training dataset sizes elevated the classification performance for all GO categories. High variance values at low training dataset sizes indicates that these models are less stable.

Table 3.4 The average prediction performance (F1-score) for GO term models belonging to different training dataset size bins. In this analysis, the training was done using only the annotations with manual experimental evidence codes.

GO categories	Performance measures (F1-score) for different training dataset sizes					
	≥ 30	≥ 100	≥ 200	≥ 300	≥ 400	≥ 500
Molecular Function	0.66	0.68	0.77	0.82	0.82	0.83
Biological Process	0.42	0.50	0.52	0.52	0.56	0.55
Cellular Component	0.50	0.59	0.64	0.63	0.64	0.65

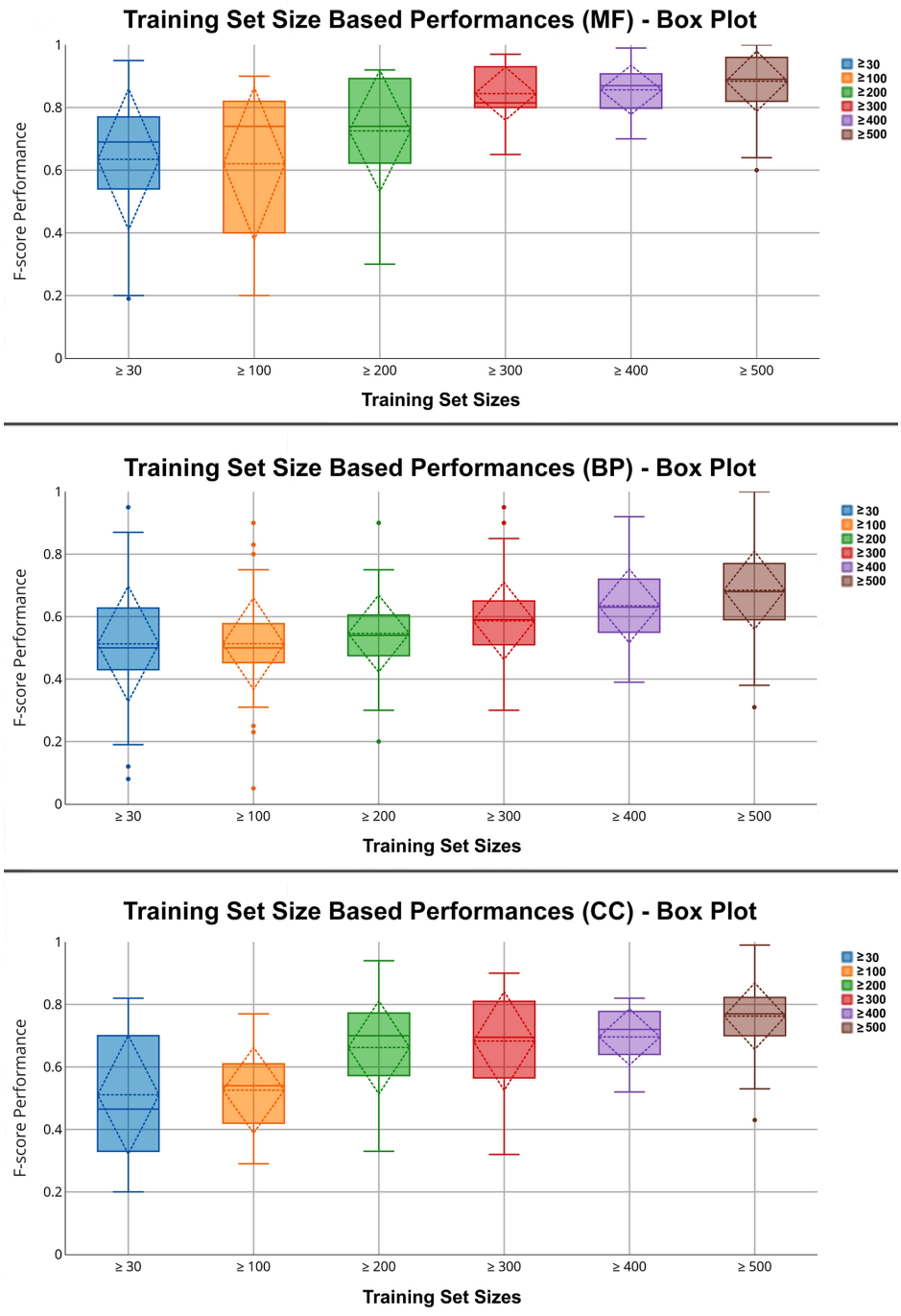


Figure 3.4. Box plots for training dataset size specific performance evaluation. Each box plot represents variance, mean and standard deviations of F1-score values (vertical axis) for models with differently sized training datasets (horizontal axis),

for each GO category. The training was done using only the annotations with manual experimental evidence codes.(Adapted from our publication [191])

3.4.4 Performance evaluation of training with electronic annotations

In DEEPred, the minimum required number of annotated proteins for each GO term (to be used in the training) is 30, which was considered as the minimum number required for statistical power. Due to this threshold, all GO terms with less than 30 annotated proteins were eliminated from the system. The eliminated terms corresponded to 25,257 out of 31,352 GO terms (31,352 is the total number of terms that have been annotated to at least one UniProtKB/Swiss-Prot protein entry with manual experimental evidence codes), which can be considered as a significant loss. The same problem exists for most of the machine learning based methods in the automated protein function prediction domain. Moreover, the DNN models with a low or moderate number of training instances (i.e., between 30 to 100 for each incorporated GO term) displayed lower performance compared to the models with high number of training samples, as discussed above. In this section, we investigated a potential way to increase the statistical power of our models by enriching the training datasets.

In the UniProtKB/SwissProt database, only 1% of the total number of GO term annotations are tagged with manual experimental evidence codes. The remaining of the GO term annotations are electronically made (evidence code: IEA), and these annotations are usually considered as less reliable due to potential errors (i.e., false positives). Normally, electronic annotations are not used for system training to avoid error propagation. In this test, we investigated the performance change when all annotations (including electronic ones) were included in the training procedure of DEEPred, and to discuss whether deep learning algorithms could handle noisy training data, as stated in the literature. For this, we calculated the predictive performances of selected models trained with all evidence code annotations and compared it with the performance of our original models.

To perform this experiment, we first identified the MF GO terms whose annotation count was increased at least four times, when electronically made annotations were incorporated. We randomly selected 25 MF GO terms that satisfied this condition, and trained/evaluated the models with 80% to 20% training-test separation, similar to our previous tests. The training dataset sizes and performance values for the "all-annotation-training" analysis are given in Table 3.5. In this table, we divided GO terms into two main categories as "previously high performance models" and "previously low performance models" based on the performances when the system was trained only with annotations of manual experimental evidence codes. The results showed that adding electronic annotations to the training procedure increased the performances of selected "previously low performance models". On the other hand, including electronic annotations in the training of "previously high-performance models" decreased their performances in some of the cases. Overall, the performance change was positive.

Table 3.5 Performance (F1-score) changes for the selected GO terms after the enrichment of training datasets with electronic annotations. In this analysis, the training was done using all of the available annotations, without any selection based on the evidence code. *NoA : Number of Annotations, ME : Manual-Experimental Evidence, AE : All Evidence.

		GO Term	NoA* (ME*)	NoA (AE*)	F1-score perf. (ME)	F1-score perf. (AE)	Perf. Change
Previously low-performance	**models**	GO:0070569	35	970	0.58	0.88	0.30
		GO:0019203	63	681	0.51	0.84	0.33
		GO:0004197	100	853	0.45	0.40	-0.05
		GO:0005524	596	85 442	0.53	0.93	0.40
		GO:0030554	689	86 319	0.51	0.90	0.39
		GO:0035639	834	98 924	0.43	0.80	0.37

Table 3.5 (continued)

	GO:0032555	951	99 286	0.51	0.89	0.38
	GO:0097367	1 395	10 413	0.41	0.73	0.32
	GO:0000166	1 487	116 408	0,53	0.82	0.29
	GO:0036094	2 059	12 634	0.40	0.72	032
	GO:0043169	2 145	119 698	0.48	0.71	0.23
	GO:0043167	4 132	20 278	0.33	0.79	0.46
Previously high-performance models	GO:0004784	38	459	0.81	0.68	-0.13
	GO:0004004	48	954	0.75	0.73	-0.02
	GO:0005525	258	14 479	0.95	0.79	-0.16
	GO:0032550	286	14 496	0.89	0.62	-0.27
	GO:0001883	289	14 506	0.89	0.87	-0.02
	GO:0032549	296	15 460	0.78	0.80	0.02
	GO:0001882	304	15 508	0.92	0.79	-0.13
	GO:0008270	520	11 385	0.83	0.71	-0.12
	GO:0032559	673	85 691	0.80	0.80	0.00
	GO:0017076	975	99 924	0.91	0.65	-0.26
	GO:0032553	1 025	100 844	0.87	0.61	-0.26
	GO:0046872	1 985	118 577	0.81	0.80	-0.01
	GO:0004784	38	459	0.81	0.68	-0.13
	Average	**861**	**47 658**	**0.68**	**0.77**	**0.10**

3.4.5 Evaluation of the Overall System Performance

Here, for the final system training we used the training dataset which was the largest one among the six different training datasets with varying sizes shown in Table 3.4 (i.e., all annotations with GO terms with at least 30 annotated proteins with manual and experimental evidence codes); this dataset was also used to measure the overall performance of DEEPred.

The overall system performance was evaluated by considering all 1,101 predictive models. For testing, the hold-out dataset (see Methods) was employed, which was not used during training. The test proteins were fed to all of the models and the system performance was calculated using precision, recall and F1-score (Table 3.6). The average prediction performance (F1-score) was calculated as 0.62, 0.46 and 0.55 for MF, BP and CC categories respectively, without using the hierarchical post-processing method (see Methods). When we employed the hierarchical post-processing procedure, which represents the finalized version of DEEPred, the overall average system performance (F1-score) was increased to 0.67, 0.51 and 0.58 for MF, BP and CC categories respectively.

Table 3.6 Performance (F1-score) changes for the selected GO terms after the enrichment of training datasets with electronic annotations. In this analysis, the training was done using all of the available annotations, without any selection based on the evidence code. *NoA : Number of Annotations, ME : Manual-Experimental Evidence, AE : All Evidence.

	without Hierarchical Post-processing			**with Hierarchical Post-processing**		
	F1-score	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**
MF	0.62	0.52	0.77	0.67	0.61	0.74
BP	0.46	0.36	0.65	0.51	0.44	0.62
CC	0.55	0.50	0.61	0.58	0.58	0.58

3.4.6 Performance Comparison Against the State-of-the-art

Two analyses were carried out for the comparison against the state-of-the-art. The first one used the CAFA2 challenge data. In CAFA2, GO term based function predictions of 126 methods from 56 research groups were evaluated. The performance results of best performing 10 methods are available in the CAFA2 report [173]. In order to yield a fair comparison with the CAFA2 participating methods, DEEPred models were re-trained using the GO annotation data from September 2013. Afterwards, DEEPred was run on the CAFA2 benchmark protein sequences and the performance results (F-max) of 0.49, 0.26, and 0.43 were obtained for MF, BP, and CC categories respectively; considering the no-knowledge benchmark set in the full evaluation mode (the official CAFA2 performance calculation parameters). For the CC category, DEEPred was among the 10 best performing methods.

Figure 3.5.A, B and C displays the 10 top performing methods in CAFA2 in terms of F-max measure along with the results of DEEPred, for the selected taxonomies where DEEPred performed well. As observed from Figure 3.5.A and B, DEEPred is among the best performers in terms of predicting MF GO terms for all prokaryotic sequences (Figure 3.5.A), specifically for *E. coli* (Figure 3.5.B). Figure 3.5.C shows that DEEPred came third in predicting BP terms for the mouse (*Mus musculus*) proteins. These results (Figure 3.5.A, B and C) also indicate that DEEPred has an added value over the conventional baseline predictors (i.e., BLAST and naive). In Figure 3.5.D, we also compared our results with the BLAST baseline classifier in terms of the GO term-centric mean area under the ROC curve (AUC) for predicting MF terms for CAFA2 benchmark sequences. As it can be seen in Figure 3.5.D, the performance of DEEPred is slightly higher than the BLAST classifier in the overall comparison considering all MF GO terms. Whereas, the performance is low for DEEPred when a comparably low number of training instances (< 1,000) of MF GO terms was used (i.e., low terms). Finally, when the MF GO terms with comparably high number of training instances (> 1,000) was employed (i.e., high terms),

DEEPred's performance surpassed BLAST. The results indicate that DEEPred is especially effective and have a significant added value over conventional methods, when the number of training instances are high.

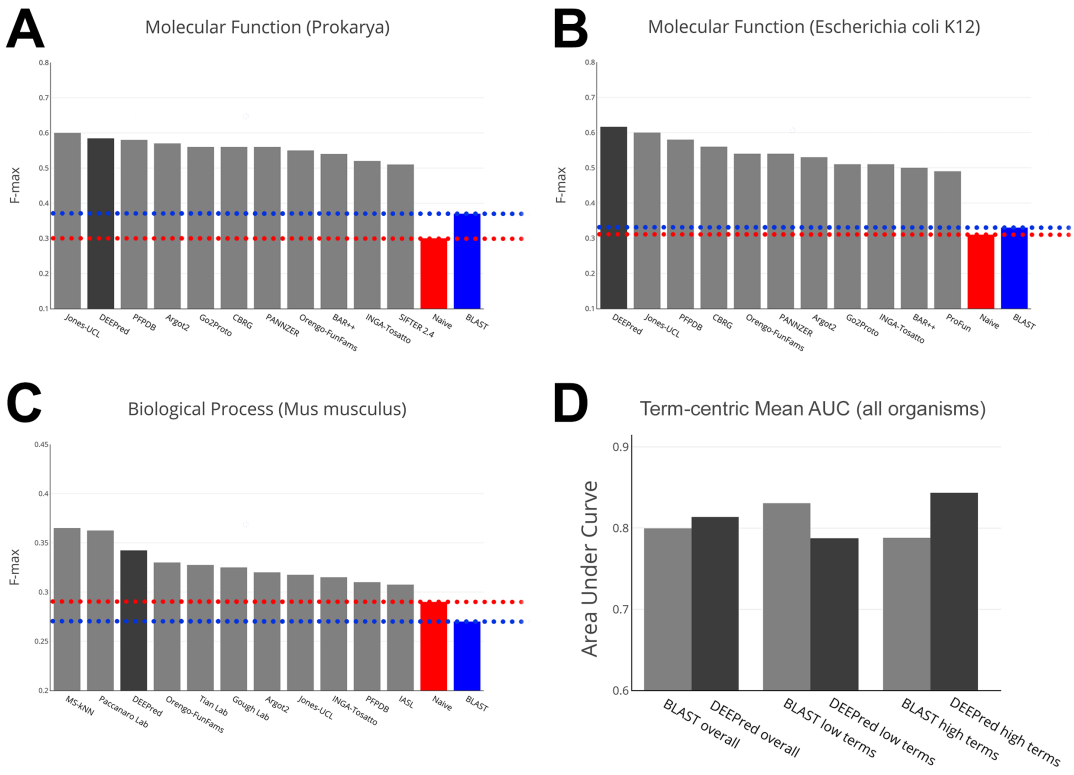


Figure 3.5. The prediction performance of DEEPred on CAFA2 challenge benchmark set. Dark gray colored bars represent the performance of DEEPred, whereas the light gray colored bars represent the state-of-the-art methods. The evaluation was carried out in the standard mode (i.e., no-knowledge benchmark sequences, the full evaluation mode), more details about the CAFA analysis can be found in CAFA GitHub repository; (A) MF term prediction performance (F-max) of top 10 CAFA participants and DEEPred on all prokaryotic benchmark sequences; (B) MF term prediction performance (F-max) of top 10 CAFA participants and DEEPred on E. coli benchmark sequences; (C) BP term prediction performance (F-max) of top 10 CAFA participants and DEEPred on mouse benchmark sequences; and (D) MF GO term-centric mean area under the ROC curve measurement comparison between BLAST and DEEPred for all MF GO terms, bars represent

terms with less than 1000 training instances (i.e., low terms) and terms with more than 1000 training instances (i.e., high terms). (Adapted from our publication [191])

The second performance analysis was done using CAFA3 challenge data, the submission period of which has ended in February 2017. The finalized benchmark dataset (protein sequences and their GO annotations) of CAFA3 was downloaded from the CAFA challenge repository on Synapse system ("benchmark20171115.tar" from https://www.synapse.org/#!Synapse:syn12278085). In total, this dataset contains 7,173 annotations (BP: 3,608, CC: 1,800 and MF: 1,765) for 3,312 proteins. The DEEPred models were re-trained using UniProt-GOA manual experimental evidence coded GO annotation data from September 2016 (the date of the official training data provided by CAFA3 organizers), and the predictions were generated for benchmark dataset protein sequences. For this analysis, we could not directly use the officially announced performance data of the top challenge performers since the results are yet to be published as of December 2018. Instead, three other sequence-based function prediction methods, namely FFPred3 [213], GoFDR [214] and DeepGO [184], were selected to be compared with DEEPred. These methods were developed and published in the last 2 years, and reported predictive performances that are better than the state-of-the-art in their own publications. For DeepGO, we downloaded the stand-alone tool, train the models with the provided training data (considering the CAFA3 submission deadline) and produced the benchmark dataset predictions. The stand-alone tool was not available for FFPred3 and GoFDR; however, CAFA3 target set function predictions were already available, as a result, we directly employed those prediction files for our analysis. We also built baseline predictors (i.e., Naïve Bayes and Blast) with CAFA3 data, as described by CAFA team. We employed performance evaluation scripts released by the CAFA team in order to calculate the performances of DEEPred, the state-of-the-art methods and the baseline classifiers. DEEPred is composed of multiple independent classifiers, each of which has its own best score threshold. For the calculation of F-max, CAFA evaluation script applies the same prediction score threshold to all predictions, which would result in the underestimation of DEEPred's performance. To avoid this, we

transformed DEEPred's prediction scores and made them comparable to each other by applying min-max normalization.

Table 3.7, Table 3.8 and Table 3.9 displays the performance results for Molecular function (MF), biological process (BP) and cellular component (CC) categories, respectively, in terms of F-max, precision, recall and Smin measures. Only the precision and recall values corresponding to the given F-max are shown. A better performance is indicated by higher F-max, precision and recall values and lower Smin values. "No-knowledge" and "All" indicates 2 different evaluation modes, where the former indicates that the methods are evaluated only using the proteins that did not have any manually curated GO annotation in the training dataset (i.e., before the challenge submission deadline), and the latter indicates that the methods are evaluated using all benchmark proteins. DEEPred was analzyed in terms of two different versions: *(i)* raw predictions coming from all predictive models, without any post-processing (i.e., DEEPred_raw), and *(ii)* finalized predictions after the hierarchical post-processing process (i.e., DEEPred_hrchy). The results are shown in Table 3.7, Table 3.8 and Table 3.9 respectively for MF, BP and CC categories of GO, where the best results for each GO category and for each performance measure is highlighted with bold font. When the second best method's performance was close to the best one, both of them are highlighted. As observed from results, the finalized version of DEEPred (i.e., DEEPred_hrchy) consistently beat the performance of the raw DEEPred predictions, indicating the effectiveness of the proposed hierarchical post-processing approach. In MF term prediction, DEEPred_hrchy shared the top place with GoFDR in terms of F-max, precision and recall (GOFDR performed slightly better in terms of Smin). Considering CC term prediction, DeepGO shared the first place with the naïve classifier, in terms of both F-max and Smin. DEEPred_hrchy was the best (in terms of F-max) for predicting BP terms of the no-knowledge proteins, and shared the first place with DeepGO and FFPred3 considering all benchmark proteins. DEEPred_hrchy was also the first in terms of Smin, for the BP category. In all GO categories, DEEPred_hrchy had a perfect precision but a low recall value, this was due to the fact that most of the prediction

scores are accumulated either close to 0 (lowest) or close to 1 (highest). During the calculation of F-max, predictions at the arbitrarily selected score thresholds are evaluated, which resulted in either a very high recall and a very low precision, or a very low recall and a very high precision for DEEPred. In CAFA3 analysis, the maximum performance of DEEPred was found around the high precision values.

Table 3.7 The prediction performance of DEEPred and the state-of-the-art protein function prediction methods on CAFA3 challenge benchmark dataset (MF Category)

	F_{max}		Precision (F_{max})		Recall (F_{max})		Smin	
	N-K	All	N-K	All	N-K	All	N-K	All
Naive	0.35	0.29	0.49	0.41	0.27	0.23	6.87	6.43
Blast	0.40	0.39	0.42	0.36	0.38	**0.44**	6.99	6.48
FFPred3	0.32	0.31	0.34	0.30	0.30	0.32	7.35	6.66
GoFDR*	**0.55**	0.45	0.67	0.55	**0.46**	0.38	**5.06**	**4.41**
DeepGO	0.40	0.34	0.58	0.48	0.30	0.27	6.36	6.01
DEEPred_raw	0.32	0.33	**1.00**	**1.00**	0.19	0.19	6.63	6.16
DEEPred_hrchy	0.49	**0.50**	**1.00**	**1.00**	0.32	0.33	5.41	5.03

3.4.7 *P. aureginosa* Case Study on biofilm formation process

We analyzed the biological relevance of the results of DEEPred over selected example predictions. For this purpose, we employed the recent CAFA PI biological process GO term assignment challenge. One of the goals in CAFA PI was the prediction of the proteins responsible for the biofilm formation (GO:0042710) process using electronically translated open reading frames (ORFs) from a specific *Pseudomonas aureginosa* strain (UCBPP-PA14) genome.

Table 3.8 The prediction performance of DEEPred and the state-of-the-art protein function prediction methods on CAFA3 challenge benchmark dataset (BP Category)

	F_{max}		Precision (F_{max})		Recall (F_{max})		Smin	
	N-K	All	N-K	All	N-K	All	N-K	All
Naive	0.26	0.30	0.25	0.39	0.26	0.24	24.27	20.85
Blast	0.28	0.32	0.22	0.27	**0.37**	0.38	25.11	21.35
FFPred3	0.26	**0.34**	0.23	0.29	0.30	**0.40**	24.74	21.48
GoFDR*	0.19	0.18	0.25	0.26	0.15	0.14	24.75	28.83
DeepGO	0.28	**0.34**	0.40	0.52	0.21	0.26	23.41	20.19
DEEPred_raw	0.16	0.16	**1.00**	**1.00**	0.09	0.09	24.65	22.05
DEEPred_hrchy	**0.32**	**0.33**	**1.00**	**1.00**	0.19	0.19	**22.04**	**19.69**

Table 3.9 The prediction performance of DEEPred and the state-of-the-art protein function prediction methods on CAFA3 challenge benchmark dataset (CC Category)

	F_{max}		Precision (F_{max})		Recall (F_{max})		Smin	
	N-K	All	N-K	All	N-K	All	N-K	All
Naive	**0.55**	**0.54**	0.56	0.58	0.55	0.50	**7.61**	**7.65**
Blast	0.46	0.45	0.39	0.39	0.56	0.53	9.74	9.94
FFPred3	0.54	0.52	0.54	0.54	0.53	0.50	8.61	8.44
GoFDR*	0.48	0.45	0.46	0.42	0.51	0.48	10.98	10.86
DeepGO	**0.54**	**0.53**	0.61	0.58	0.48	0.49	**7.68**	**7.55**
DEEPred_raw	0.30	0.29	0.19	0.18	**0.69**	**0.69**	10.68	10.41
DEEPred_hrchy	0.34	0.35	**1.00**	**1.00**	0.20	0.22	9.85	9.53

Pseudomonas aureginosa is a gram-negative pathogenic bacteria with high medical importance due to its ability to cause infection in human (e.g., pneumonia) and its highly effective antibiotic resistance mechanisms [215]. An important factor contributing to the infectious capabilities of various bacterial and fungal species is their ability to form biofilms. A biofilm is a matrix layer made up of extracellular polymers and the microorganisms themselves. Biofilms adhere to solid surfaces and provide a medium for the cells to proliferate and resistance to environmental stress. Due to this reason, understanding the biofilm formation mechanisms in pathogenic microorganisms have high importance [216], [217].

In order to annotate ORF sequences from *P. aureginosa* UCBPP-PA14 strain with biofilm formation GO term using DEEPred, we generated a single task feed-forward DNN model. The reason behind not using a multi-task model here was to prevent the potential effect of the selection of the accompanying GO terms to the predictive performance. The positive training dataset for this model was generated from all UniProtKB/Swiss-Prot protein records that were annotated either with the corresponding GO term or with its descendants with manual and experimental evidence codes, yielding 254 proteins. The negative training dataset was selected from the protein entries that were neither annotated with the corresponding GO term nor any of its descendants (the same number of samples were selected randomly to match the positive training dataset). The model was trained by optimizing the hyper-parameters and its performance was measured via 5-fold cross validation. The performance results in terms of precision, recall and F1-score were 0.71, 0.84 and 0.77 respectively. The finalized models were then employed to predict functions for CAFA PI *P. aureginosa* ORF targets.

From a literature review, we identified 8 genes (wspA, wspR, rocR, yfiN, tpbB, fleQ, fimX and PA2572) in the *P. aureginosa* reference genome that are associated with biofilm formation, but not annotated with the corresponding GO term or its functionally related neighboring terms, in the source databases at the time of this analysis (as a result, they are not presented in our training dataset). Out of these 8 genes/proteins wspR, yfiN, tpbB and fimX contain the GGDEF domain, which is

responsible for synthesizing cyclic di-GMP and thus take part in the biofilm formation process [218]. Two of these genes/proteins, yfiN and tpbB, additionally contain the CHASE8 sensor domain, which controls the levels of extracellular DNA and regulates biofilm formation [219]. The mechanism by which these 8 genes/proteins contribute to the formation of biofilm are explained in two articles by Cheng [220] and Ryan *et al.* [221].We obtained the protein sequences of these genes from the UniProt database, then aligned them to the CAFA PI *P. aureginosa* UCBPP-PA14 strain's target ORF sequences to identify the CAFA PI target sequences corresponding to these genes with a cut-off of greater than 98% identity. The reason behind this application was that the CAFA PI target dataset ORF sequences were unknown. Finally, we analyzed the equivalent *P. aureginosa* ORFs of these 8 genes in the target dataset using DEEPred's biofilm formation process model and examined the prediction scores.

Table 3.10 displays the gene symbols, protein (UniProt) accessions and biofilm formation GO term prediction scores produced by DEEPred for the selected genes/proteins. As observed in Table 3.10, 4 out of 8 genes/proteins (i.e., gene symbols: wspA, wspR, rocR and PA2572) received high prediction scores for the biofilm production term and thus successfully identified by DEEPred. Two genes/proteins (i.e., gene symbols: yfiN and tpbB) received moderate scores, which were still sufficient to produce a prediction. The remaining two genes/proteins (i.e., gene symbols: fleQ and fimX) could not be associated with the corresponding GO term at all. We also carried out a BLAST search in order to observe if these predictions could be produced by a conventional sequence similarity search. For this, the amino acid sequence of each of the 8 genes/proteins was searched against the whole UniProtKB with an e-value threshold of 100. The BLAST search revealed that none of the best 1,000 BLAST hits (50% or greater identity) possessed the biofilm formation GO term or any of its ancestor or descendant terms as annotations, and thus BLAST failed to annotate these genes/proteins. Since none of these 8 genes/proteins (or their BLAST hits) have been annotated with a GO term related to the biofilm formation function on the GO DAG; there were no protein sequences in

the training dataset of DEEPred that were similar to these genes. As a result, the accurate predictions cannot be the result of a simple annotation transfer between close homologs.

Table 3.10 DEEPred's biofilm formation term (GO:0042710) prediction results for the selected *P. aureginosa* proteins.

Gene symbol	Protein accession (UniProt)	DEEPred prediction score
wspA	A0A0H2ZEY3	0.99
rocR	A0A0C7D525	0.98
PA2572	Q9I0R4	0.98
wspR	A0A0H2ZEX4	0.95
yfiN	A0A0C7ADU5	0.68
tpbB	Q9I4L5	0.68
fleQ	A0A0H2Z7X4	0.05
fimX	A0A0H2ZHA6	0.02

3.4.8 Participation of CAFA PI Challenge and Ranking

We also participated CAFA PI challenge with DEEPred method where the objective was to provide of *term-centric* predictions for a given set of protein sequences coming from two species which are *P. aeruginosa* and *Candida albicans* [222]. Participants were asked to provide funtional predictions for two functions which are *motility (GO:0001539)* and *biofilm formation (GO:0042710)* for 5,892 *P. aeruginosa* proteins and only *biofilm formation (GO:0042710)* predictions for 12,421 proteins from *Candida albicans*. 49 participants (25 teams and 24 individual participants) were registered for CAFA PI challenge and each team or individual researchers were allowed to submit up to 3 independent predictions. The prediction results were evaluated based on genome-wide screening experiments which were

performed to identify the genes that are associated with *biofilm formation* and *motility* functions [222]. The experiments were performed by experimental biologists from Dartmouth College. 403 genes from *P. aeruginosa* were associated with the *motility* function based on screening experiments. The performances of the participants were evaluated based on the data coming from novel experimental results. Our team (METU-CanSyL) ranked fourth among all participants in motility category [222]. The results of the top performing five teams are given in Figure 3.6.

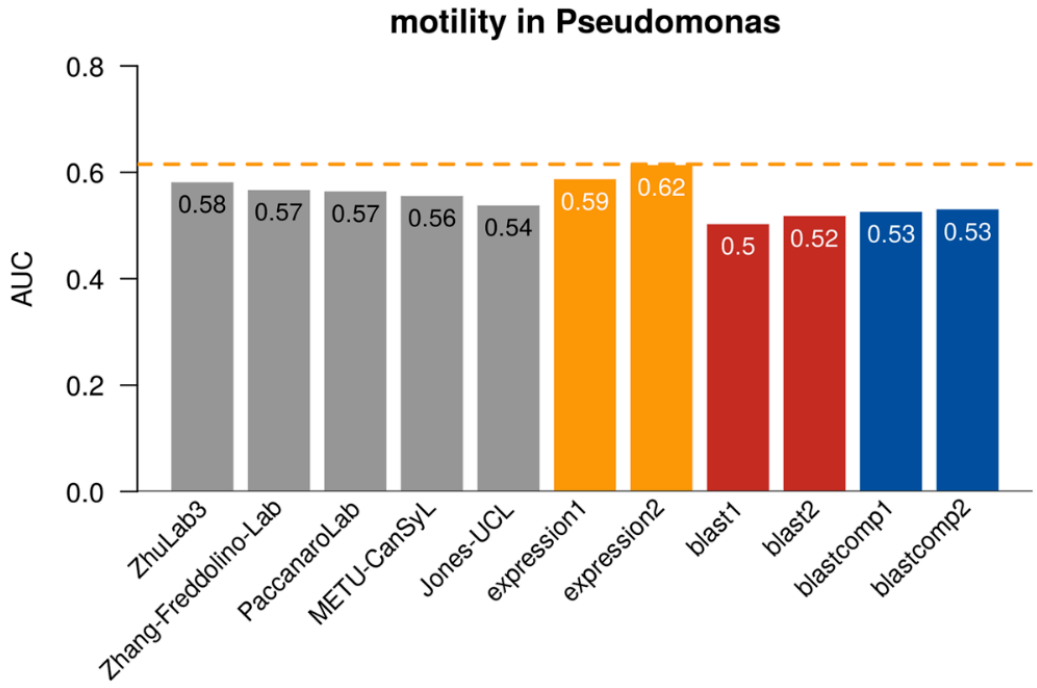


Figure 3.6. CAFA PI Top 5 best performed method based on Area Under Curve (AUC) results. Our team (METU-CanSyL) ranked fourth among all teams. (Adapted from CAFA 3/PI publication [222])

3.5 Discussion and Conclusion

Deep learning algorithms have shown to significantly enhance the classification performances in various fields; however, it was not thoroughly investigated in terms of their applications to the protein function prediction area at large-scale. In this study, we described the DEEPred method for predicting GO term based protein

functions using a stack of feed-forward multi-task deep neural networks. As input, DEEPred only requires the amino acid sequences of proteins. We carried out several tests to investigate the behavior of DNN-based models in protein function prediction. The input feature type selection test revealed that our in-house protein descriptor SPMap had a better performance compared to the conventional conjoint triad and pseudo-amino acid composition features. However, this performance increase comes with a cost in terms of higher vector dimensionality (i.e., SPMap has between 1000 to 2000-dimensions as opposed to 373 for conjoint triad and 50 for pseudo-amino acid composition), which elevates the computational complexity. It would also be interesting to analyze additional protein feature types, especially the amino acid descriptors frequently used in protein-ligand binding prediction studies [223].

The reasons behind choosing DEEPred's specific DNN architecture was first, this is a basic form and thus it is straightforward to train and apply. In other words, it requires minimal amount manual design work compared to specialized complex networks such as the Inception Network [224]. This is especially important considering the fact that more than one thousand independent networks should have been trained. Second, computational resources required to train this architecture is lower compared to, again, the complex networks.

In DEEPred, we considered multi-task DNNs (as opposed to single-task DNNs) due to various advantages attributed to multi-task networks such as: *(i)* the ability to share knowledge between tasks; which supports the system in the case where there are a low number of training instances and *(ii)* training with less models to improve the training run times. However, multi-task DNNs also have disadvantages especially when the high number of tasks compels the generation of multiple models. The problem here is the efficient grouping of the tasks (i.e., GO terms in our case) so that the tasks under a model would become alternatives (i.e. orthogonal) to each other. We tried to achieve this by first, grouping GO terms from the same level and second, where possible placing the sibling terms together under the same model. In most cases, it was not possible to find a sufficient number of sibling terms and thus semantically unrelated terms from the same level ended up in the same model.

Nevertheless, this was not a crucial problem since it is possible for a multi-functional query protein to receive high prediction scores for multiple GO terms under the same model. Another important point during the term grouping was placing GO terms with similar number of annotated proteins under the same group. According to our observations, models containing tasks with highly unbalanced number of training instances perform poorly (this is also one of the reasons why generating only one model to predict all GO terms would be a poor design choice). Due to these reasons, generating the models required a considerable amount of manual work, none of which would be required if we employed single-task networks. It would also be possible to achieve higher performance values with single-task DNNs, especially where there is sufficient number of training instances. We did not consider single-task networks mainly because it is not feasible to train tens of thousands of networks (when the hyper-parameter optimization step is considered the number would increase to billions of training jobs) to cover the whole functional space. In the future, it would be interesting to see algorithmic solutions to the feasibility problems related to single-task networks. With such solutions we could construct and test a single-task DNN-based system for protein function prediction.

In this study, we trained several DNN models using 6 different groups of training datasets containing GO terms with differing number of training samples to investigate the performance changes due to changes in the training sample size. Our training dataset size performance evaluation results showed that there is a general trend of performance increase with the increasing number of training samples, which means that including GO terms with a small number of protein associations into models decreased the overall performance. Therefore, our findings are in accordance with the literature regarding training data sizes being one of the key factors that affect the predictive performance of deep learning algorithms; though, the research community started to focus on developing novel deep learning based approaches to address training dataset size related problems [135].

In this work, we also investigated if there is a relationship between levels of GO terms on the GO DAG and the classification performances and we found out that

there is no such correlation. In addition, we observed that the variance in performance between different GO levels decreases as the training dataset size increases for molecular function and cellular component categories. For the biological process category, the overall performance increases with increasing GO training dataset sizes, however the variance is relatively higher. The main reason behind this may be attributed to the biological process GO terms representing complex processes (e.g., GO:0006099 - tricarboxylic acid cycle) that involves several molecular events, which is hard to associate with a sequence signature. The results also showed that performance variance of cellular component GO terms is lower compared to the molecular function and biological process categories. The reason for such observation could be that the hierarchy between cellular compartment GO terms is inherently available within cells, which results in better defined hierarchical relationships between cellular component GO terms.

In most of the protein function prediction methods, training was performed using only the annotations with manual and experimental evidence codes. The disadvantage of this approach is that most GO terms are left with a small number of annotated proteins, which is usually not sufficient for a machine learning model training. Therefore, the functions defined by these terms cannot be predicted efficiently. One solution would be to include the annotations with non-experimental evidence codes such as the electronic annotations (i.e., the annotations produced by other automated approaches). For example, the number of MF GO terms that have more than 30 protein associations is calculated as 911 when we only considered the annotations with manual experimental evidence codes. However, when we considered the annotations with all evidence codes, this number increases to 2,776, meaning that, if the annotations with all evidences are included, it is possible to provide predictions for significantly more GO terms. The main downside of adding annotations with non-manual/experimental evidence codes to the training dataset is the false positive samples, which would result in error propagation. Another potential limitation of this application would be that the predictive performance of the models with training datasets dominated by the electronic annotations would still be low

(even though the number of training instances are increased), due to the fact that the sequences of most of the electronically annotated genes/proteins under a distinct GO term would be extremely similar to each other; and thus, would not provide the required sample diversity.

To investigate this issue of training dataset enrichment, we observed a performance change when the annotations with all evidence codes were included in the training and compared the performance results (Table 3.5). The evaluation results showed that the performance of the previously low performance GO term models were increased significantly, which indicates that deep learning algorithms are tolerant to noise in the learning data. Therefore, annotations with less reliable evidence codes can be included in the training of low performed models, where there is still room for significant performance improvement. However, including less reliable annotations in the training dataset of previously high performance models decreased the performance for more than half of them.

In DEEPred, we employed a hierarchical post-processing method (in order to avoid false positive hits) by taking the prediction scores of the parents of the target GO term into account, along with the score of the target term. The evaluation results indicated that the recall values were slightly decreased and the precision scores were noticeably increased when we employed the hierarchical post-processing procedure, producing an increased overall performance in terms of F1-score (Table 3.6). In this setting, the resulting predictions can be considered more reliable. This is also indicated by the improved F-max values at the CAFA3 benchmark test (Table 3.7, 3.8 and 3.9).

In our performance tests, DEEPred performed slightly better than the state-of-the-art methods in some cases, and produced roughly similar results in others. However, we did not observe an unprecedented performance increase; probably because we did not focus on specific functional families to optimize the system performance. Instead, we investigated the applicability of DNNs for constructing large-scale automated protein function prediction pipelines. We believe that this investigation

will be valuable for computational scientists in terms of developing DNN-based biological prediction methods. According to our observations, it is feasible to use DNNs in large-scale biological data analysis pipelines, where it may be possible to achieve performances higher than the state-of-the-art methods with additional optimization. However, feed-forward DNN based modeling is probably not a good choice for the functional terms with low or moderate number of annotated proteins (at least without a pre-processing step such as the training dataset enrichment), for which conventional machine learning solutions or DNN-based methods specialized in low-data training may be considered.

Generally, function prediction methods that incorporate multiple types of protein features at once (e.g., sequence, protein-protein interactions - PPIs, 3-D structures and annotations and etc.) perform better compared to methods that incorporate sequences, solely [173], [174]. However, there are two main disadvantages of this approach. First of all, query proteins are required to have a substantial amount of characterization (especially in terms of PPIs and 3-D structures) in order for these methods to accept them as queries. Structurally well characterized proteins usually have a high-quality functional annotations, thus, function prediction methods are not required in the first place. Second, running times of these methods are generally multiple orders of magnitude higher compared to the sequence-based predictors, which significantly hinders their large-scale use such as the analysis of newly sequenced genomes.

Finally, we carried out a case study to discuss the biological relevance of the results produced by DEEPred, by predicting the *Pseudomonas aureginosa* ORF sequences that take part in the biofilm formation biological process. DEEPred managed to identify 6 out of 8 proteins that are reported to play roles in the biofilm formation process, which are not yet annotated with the corresponding GO term (or any of its descendent terms) in the source biological databases as of April 2018. As a result, it can be said that without any prior knowledge DEEPred produced biologically relevant predictions considering the selected process. It is also evident that DEEPred performed significantly better in this test, compared to the baseline classifier (i.e.,

BLAST). It is difficult to identify how deep neural networks managed to annotate these proteins where BLAST failed significantly; however, it can be attributed to DNN's ability to extract signatures (relevant to the task at hand) hidden in the sequences, by consequent levels of data abstraction.

The methodological approach proposed in this study can easily be translated into the prediction of various types of biomolecular ontologies/attributes (e.g., protein families, interactions, pathways, subcellular locations, catalytic activities, EC numbers and structural features) and biomedical entity associations (e.g., gene-phenotype-disease relations and drug-target interactions).

CHAPTER 4

DEEPScreen: DRUG-TARGET INTERACTION PREDICTION WITH CONVOLUTIONAL NEURAL NETWORKS USING 2-D STRUCTURAL COMPOUND REPRESENTATIONS

4.1 Chapter Overview

[4]The identification of physical interactions between drug candidate compounds and target biomolecules is an important process in drug discovery. Since conventional screening procedures are expensive and time consuming, computational approaches are employed to provide aid by automatically predicting novel drug–target interactions (DTIs). In this study, we propose a large-scale DTI prediction system, DEEPScreen, for early stage drug discovery, using deep convolutional neural networks. One of the main advantages of DEEPScreen is employing readily available 2-D structural representations of compounds at the input level instead of conventional descriptors that display limited performance. DEEPScreen learns complex features inherently from the 2-D representations, thus producing highly accurate predictions. The DEEPScreen system was trained for 704 target proteins (using curated bioactivity data) and finalized with rigorous hyper-parameter optimization tests. We compared the performance of DEEPScreen against the state-of-the-art on multiple benchmark datasets to indicate the effectiveness of the proposed approach and verified selected novel predictions through molecular docking analysis and literature-based validation. Finally, JAK proteins that were predicted by DEEPScreen as new targets of a well-known drug cladribine were

[4] The content of this chapter was published in *Chemical Science* journal in 2020 [266]. Please note that only the parts that I worked on were included from our publication.

experimentally demonstrated *in vitro* on cancer cells through STAT3 phosphorylation, which is the downstream effector protein. The DEEPScreen system can be exploited in the fields of drug discovery and repurposing for *in silico* screening of the chemogenomic space, to provide novel DTIs which can be experimentally pursued. The source code, trained "ready-to-use" prediction models, all datasets and the results of this study are available at https://github.com/cansyl/DEEPScreen.

This chapter consists of the parts that I mainly worked in DeepScreen. The rest of the conducted research and analysis can be reached from our publication. My specific contributions in DeepScreen are listed below:

- Development and design of the proposed method;
- Implementation of the overall system ;
- Investigation of ChEMBL database and defining preprocessing and filtering rules to create reliable dataset;
- Investigation of state-of-the-art methods and benchmarking datasets;
- Implementation of scripts for the analysis and discussions.

4.2 Introduction

One of the initial steps of drug discovery is the identification of novel drug-like compounds that interact with the predefined target proteins. *In vitro* / *In vivo* and high-throughput screening experiments are performed to detect novel compounds with the desired interactive properties. However, high costs and temporal requirements makes it infeasible to scan massive target and compound spaces[9]. Due to this reason, the rate of the identification of novel drugs has substantially been decreased [10]. Currently, there are more than 90 million drug candidate compound records in compound and bioactivity databases such as ChEMBL [225] and PubChem [226] (combined), whereas, the size estimation for the whole “drug-like” chemical space is around 10^{60} [227]. On the other hand, the current number of drugs

(FDA approved or at the experimental stage) is around 10,000, according to DrugBank [141]. In addition, out of the 20,000 proteins in the human proteome, less than 3,000 of them are targeted by known drugs [1], [11]. As the statistics indicates, the current knowledge about the drug-target space is limited, and novel approaches are required to widen our knowledge.

The studies published so far have indicated that DTI prediction is an open problem, where not only novel ML algorithms but also new data representation approaches are required to shed light on the un-charted parts of the DTI space [89], [122], [232]–[234], [123], [134], [135], [223], [228]–[231], and for other related tasks such as reaction [235] and reactivity predictions [236] and de novo molecular design [237], [238]. This effort comprises the identification of novel drug candidate compounds, as well as the repurposing of the existing drugs on the market [239]. Additionally, in order for the DTI prediction methods to be useful in real-world drug discovery and development research, they should be made available to the research community as tools and/or services via open access repositories. Some examples to the available deep learning based frameworks and tools in the literature for various purposes in computational chemistry based drug discovery can be given as: gnina, a DL framework for molecular docking (repository: https://github.com/gnina/gnina) [240]–[243]; Chainer Chemistry, a DL framework for chemical property prediction, based on Chainer (repository: https://github.com/chainer/chainer-chemistry) [244]; DeepChem, a comprehensive open-source toolchain for DL in drug discovery (repository: https://github.com/deepchem/deepchem) [245]; MoleculeNet, a benchmarking system for Molecular Machine Learning, which builds on DeepChem (repository: http://moleculenet.ai/) [234]; and SELFIES, a sequence-based representation of semantically constrained graphs, which is applicable to represent chemical compound structures as graphs (repository: https://github.com/aspuru-guzik-group/selfies) [246].

In this study, we propose DEEPScreen, a deep convolutional neural network (DCNN) based DTI prediction system that utilizes readily available 2-D structural compound representations as input features, instead of using conventional

descriptors such as the molecular fingerprints [247]. The main advantage of DEEPScreen is increasing the DTI prediction performances with the use of 2-D compound images, that is assumed to have a higher coverage in terms of compound features, compared to the conventional featurization approaches (e.g., fingerprints), which have issues related to generalization over the whole DTI space [134], [248]. DEEPScreen system's high-performance DCNNs inherently learn these complex features from the 2-D structural drawings, to produce highly accurate novel DTI predictions at large scale. Image-based representations of drugs and drug candidate compounds reflect the natural molecular state of these small molecules (i.e., atoms and bonds), which also contain the features/properties determining their physical interactions with the intended targets. Recently, image-based or similar structural representations of compounds have been incorporated as input for predictive tasks under different contexts (e.g., toxicity, solubility, and other selected biochemical and physical properties) in the general field of drug discovery and development [248]–[251], but have not been investigated in terms of the binary prediction of physical interactions between target proteins and drug candidate compounds, which is one of the fundamental steps in early drug discovery. In this work, we aimed to provide such investigation, and as output, we propose a highly-optimised and practical DTI prediction system that covers a significant portion of the known bio-interaction space, with a performance that surpasses the state-of-the-art.

The proposed system, DEEPScreen, is composed of 704 predictive models, each one is independently optimized to accurately predict interacting small molecule ligands for a unique target protein. DEEPScreen has been validated and tested using various benchmarking datasets, and compared with the state-of-the-art DTI predictors using both conventional and deep ML models. Additionally, DEEPScreen target models were run on more than a million compound records in ChEMBL database to produce large-scale novel DTIs. We also validated selected novel predictions using three different approaches: *(i)* from the literature, in terms of drug repurposing, *(ii)* with computational structural docking analysis, and *(iii)* via *in vitro* wet-lab experiments. Finally, we constructed DEEPScreen as a ready to use collection of predictive

models and made it available through an open access repository together with all of the datasets and the results of the study at: https://github.com/cansyl/DEEPScreen.

4.3 Methods

4.3.1 Generation of the Fundamental Training Dataset

ChEMBL database (v23) was employed to create the training dataset of DEEPScreen. There are 14,675,320 data points (i.e., DTIs) in ChEMBL v23. We applied several filtering and pre-processing steps on this data to create a reliable training dataset. First of all, data points were filtered with respect to "target type" (i.e., single protein), "taxonomy" (i.e., human and selected model organisms), "assay type" (i.e., binding and functional assays) and "standard type" (i.e., IC50, EC50, AC50, Ki, Kd and Potency) attributes, which reduced the set to 3,919,275 data points. We observed that there were duplicate measurements inside this dataset that are coming from different bioassays (i.e., 879,848 of the bioactivity data points belonged to 374,024 unique drug-target pairs). To handle these cases, we identified the median bioactivity value for each pair and assigned this value as the sole bioactivity measurement. At the end of this application, 3,413,451 bioactivity measurements were left. This dataset contained data points from both binding and functional assays. In order to further eliminate a potential ambiguity considering the physical binding of the compounds to their targets, we discarded the functional assays and kept the binding assays with an additional filtering on "assay type". Finally, we removed the bioactivity measurements without a pChEMBL value, which is used to obtain comparable measures of half-maximal response on a negative logarithmic scale in ChEMBL. The presence of a pChEMBL value for a data point indicates that the corresponding record has been curated, and thus, reliable. After the abovementioned processing steps, the number of bioactivity points were 769,935.

Subsequently, we constructed positive (active) and negative (inactive) training datasets as follows: For each target, compounds with bioactivity values ≤ 10 μM

were selected as positive training samples and compounds with bioactivity values ≥ 20 μM were selected as negative samples. In DEEPScreen, only the target proteins with at least 100 active ligands were modelled, in order not to lose the statistical power. This application provided models for 704 target proteins from multiple highly studied organisms. These organisms, together with the distribution of target proteins for each organism are: Homo sapiens (human): 523, Rattus norvegicus (rat): 88, Mus musculus (mouse): 34, Bos taurus (Bovine): 22, Cavia porcellus (Guinea pig): 13, Sus scrofa (Pig): 9, Oryctolagus cuniculus (Rabbit): 5, Canis familiaris (dog): 3, Equus caballus (horse): 2, Ovis aries (Sheep): 2, Cricetulus griseus (Chinese hamster): 1, Mesocricetus auratus (Golden hamster): 1 and Macaca mulatta (Rhesus macaque): 1. The UniProt accessions, encoding gene names, ChEMBL ids and the taxonomic information of these proteins are given in the repository of DEEPScreen. Each target's training set contained a mixture of activity measurements with roughly comparable standard types (e.g., IC50, EC50, AC50, Ki, Kd and Potency).

The selection procedure explained above generated positive and negative training datasets with varying sizes for each target. In order to balance the positive and negative datasets, we selected negative samples equal to the number of positive instances. However, for many targets, the number of negative points were lower than the positives. In these cases, we applied a target similarity-based inactive dataset enrichment method to populate the negative training sets (instead of randomly selecting compounds), using the idea of similar targets have similar actives and inactives. For this, we first calculated pairwise similarities between all target proteins within a BLAST search. For each target having insufficient number of inactive compounds, we sorted all remaining target proteins with descending sequence similarity. Then, starting from the top of the list, we populated the inactive dataset of the corresponding target using the known inactive compounds of similar targets, until the active and inactive datasets are balanced. We applied 20% sequence similarity threshold, meaning that we did not consider the inactives of targets, whose sequence similarity to the query protein is less than 20%. The finalized training dataset for 704 target proteins contained 412,347 active data points (≤ 10 μM) and

377,090 inactive data points (≥ 20 μM). Before the negative dataset enrichment procedure, the total number of inactive instances for 704 targets were only 35,567. Both the pre-processed ChEMBL dataset (769,935 data points) and the finalized active and inactive training datasets for 704 targets are provided in the repository of DEEPScreen. We believe the resulting bioactivity dataset is reliable, and it can be used as standard training/test sets in future DTI prediction studies. The training data filtering and pre-processing operations are shown in Figure 4.1.

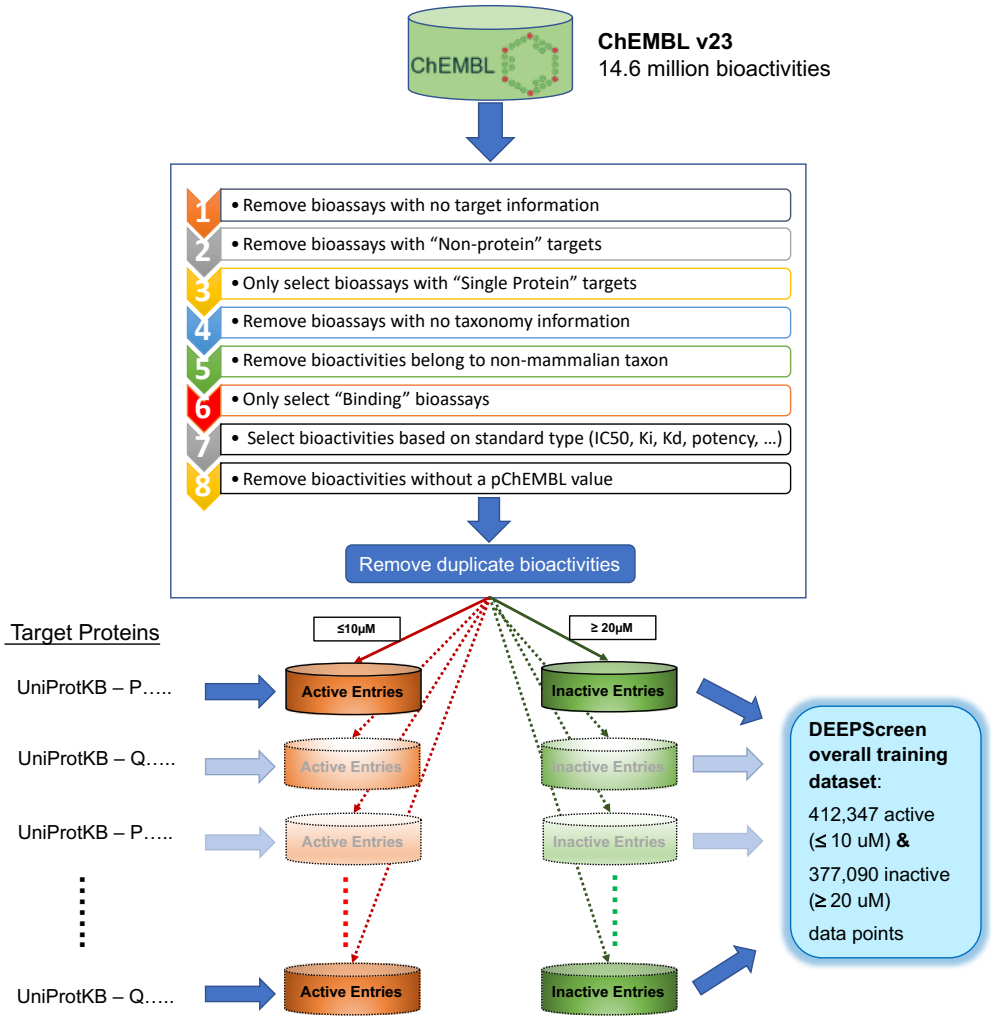


Figure 4.1. Data filtering and processing steps to create the training dataset of each target protein model. Predictive models were trained for 704 target proteins, each of which have at least 100 known active ligands in the ChEMBL database.

4.3.2 Representation of Input Samples and the Generation of Feature Vectors

In DEEPScreen system, each compound is represented by a 200-by-200 pixel 2-D image displaying the molecular structure (i.e., skeletal formula). Although 2-D compound images are readily available in different chemical and bioactivity databases, there is no standardization in terms of the representation of atoms/bonds, functional groups and the stereochemistry. Due to this reason, we employed SMILES strings of compounds to generate the 2-D structural images, since SMILES is a standard representation that can be found in open access bioactivity data repositories, which contain the whole information required to generate the 2-D images. We employed the RDkit tool Python package (v2016.09.4) for image generation [252]. A few examples from the generated images are shown in Figure 4.2.

2-D images generated by RDkit are reported to have standard and unique representation, which is achieved by applying a canonical orientation in all cases [253]. There are special cases, which are not handled well, such as the stereochemistry. However, this problem is not related to the generation of 2-D images by RDkit, but to the SMILES representations being non-stereospecific. In this study, we omitted stereochemistry since the cases correspond an insignificant portion of the whole ChEMBL database [254].

We carried out a small scale analysis to determine the input image size of the DEEPScreen system. We selected 100-by-100, 200-by-200 and 400-by-400 pixel image sizes for the test (sizes smaller than 100-by-100 were inadequate to draw molecules and sizes higher than 400-by-400 were too large to train the system with due to increased complexity). We generated the training and test compound images with the selected sizes for 3 target proteins: Muscarinic acetylcholine receptor M5 (CHRM5) - CHEMBL2035, Carbonic anhydrase VB (CA5B) - CHEMBL3969 and Renin - CHEMBL286. After that, we trained 9 models (3 targets for 3 different images sizes) and optimized the hyper-parameters with grid-search. The finalized models were subjected to performance analysis by querying the test dataset

compounds. We also recorded the average computational parameters in terms of run time and memory (the same amount of CPU power have been used for each model train/test run). The test results are given in Table 4.1.

Table 4.1 The performance results and the computational requirements (in training) of 3 target protein models in the input image size analysis.

Test performance results of the best model (average of 3 target protein models):	**Input Image Size**		
	100x100	200x200	400x400
MCC	0.59	0.69	0.65
F1-score	0.79	0.84	0.83
Accuracy	0.79	0.84	0.83
Precision	0.83	0.87	0.84
Recall	0.76	0.83	0.83
Computational requirements for in-house DCNN and Inception model training (average of 3 target protein models):			
CNNModel run time (min)	8	46	192
Inception run time (min)	75	470	-
CNNModel memory (Gb)	0.7	2.6	7.9
Inception memory (Gb)	3.3	7.3	-

As shown in Table 4.1, the average predictive performance (in terms of MCC) significantly increased by 17% when the input image size is changed from 100-by-100 to 200-by-200. A similar performance increase was not observed when the input image size is changed from 200-by-200 to 400-by-400. Considering the run times, there was a significant increase both between 100-by-100 and 200-by-200; and 200-by-200 and 400-by-400. The run times for DCNN models were acceptable; however, it was not feasible to train the Inception model with 400-by-400 due to extremely long run times. Considering the performance results along with the computational requirements, 400-by-400 was found to be non-feasible. Finally, for memory

requirements, again the results were reasonable for DCNN models and for Inception models when the image sizes are either 100-by-100 or 200-by-200. These results indicated that, the best performances were achieved with 200-by-200 image sizes, with reasonable computational requirements. As a result, 200-by-200 image size was chosen as default for the DEEPScreen system. Moreover, we observed in several cases that the size 100-by-100 was not sufficient to express large compounds properly. The whole image size analysis results are given in the repository of the study.

4.3.3 Neural Network Architecture of DEEPScreen

Deep convolutional neural networks are a specialized group of artificial neural networks consisting of alternating, convolution and pooling layers, which extracts features automatically [115], [255]. DCNNs have been dominating the image processing area in the last few years, achieving significantly higher performances compared to the state-of-the-art of the time [115], [224], [256]. DCNNs run a small window over the input feature vector at both training and test phases as a feature detector and learn various features from the input regardless of their absolute position within the input feature vector. Convolution layers compute the dot product between the entries of the filter and the input, producing an activation map of that filter. Suppose that the size of the layer, on which the convolution operation is to be performed (layer #: $l-1$) is NxN and the following convolutional layer has the layer #: l. Then, the value of the unit x_{ij} in the l^{th} layer, x_{ij}^{l}, is calculated by the convolution operation (assuming no padding and stride of 1) using the following equation:

$$x_{ij}^{l} = \sum_{a=0}^{f-1} \sum_{b=0}^{f-1} w_{ab}\, y_{(i+a)(j+b)}^{l-1}$$

In the equation above, f stands for filter size, w stands for fxf filter and y_{ij}^{l-1}stands for the value of the i^{th} row and j^{th} column in the $(l-1)^{\text{th}}$ layer. Subsequently, a non-linear function σ such as the rectified linear unit (ReLU) is applied to x_{ij}^{l}:

$$y_{ij}^{l} = \sigma(x_{ij}^{l})$$

At the end of the convolution operation, the size of the l^{th} layer becomes $(N-f+1)x(N-f+1)$. The parameters of the networks are optimized during the backpropagation step, by minimizing the following cross-entropy loss function:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{K}\sum_{i}^{K}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]$$

In the equation above, $\hat{y}$ stands for prediction score, y stands for actual label and K stands for number of examples in mini-batches. Although the most standard form of DCNNs employ 2-D convolutions, 1-D or 3-D convolutions are possible.

Pooling layers combine the output of neuron clusters at one layer into a single neuron in the subsequent layer (i.e., down-sampling) with the aim of reducing the number of parameters and the computational work, and to prevent overfitting. Max pooling layer is commonly used in DCNNs and it works by running a window sequentially and taking the maximum of the region covered by the window, where each maximum value becomes a unit in the next layer. One of the most popular and widely used regularization techniques is dropout [193]. This method randomly deactivates some of the neurons at various layers along with their connections at every epoch during the training procedure. By this, the system prevents overfitting, thus, the constructed models are more generalized.

In this study, we considered the DTI prediction as a binary classification problem, where the output can either be positive (i.e., active, interacting or "1") or negative (i.e., inactive, non-interacting or "0"), referring to the relation between the query compound and the modelled target protein. For this purpose, an individual model was created for each target protein (i.e., the single task approach). In terms of the employed DCNN architectures, we initially chose 3 options: Inception [224], AlexNET [256]; and an in-house built DCNN architecture. AlexNET architecture is a DCNN with stacked convolutional layers. It contains 5 convolutional and 3 fully connected layers. Inception is a highly specialized DCNN architecture. In standard DCNNs, filters with a uniform size are used in each level of convolutional layers, whereas in Inception, multiple filters with different sizes are combined in the same level (i.e., Inception modules), to be able capture highly complex features. Various combinations of Inception modules are designed to create extremely deep and wide networks to achieve high predictive performance in practical training run times. Detailed information about the Inception network can be found in Szegedy *et al*. [224]. Both AlexNET and Inception displayed top performances in image classification tasks [224], [256]. For our in-house designed DCNN models, we used a simpler architecture (compared to Inception), which is composed of 5 convolutional + pooling and 1 fully-connected layer preceding the output layer. Each convolutional layer was followed by a ReLU activation function and max pooling layers. The last convolutional layer is flattened and connected to a fully-connected layer, followed by the output layer. We used Softmax activation function in the output layer. A generic representation of the constructed DCNN models is given in Figure 4.2. TFLearn framework version 0.3.2, cairosvg 2.1.2, rdkit 2016.09.4 were employed for the construction of the DEEPScreen system [257].

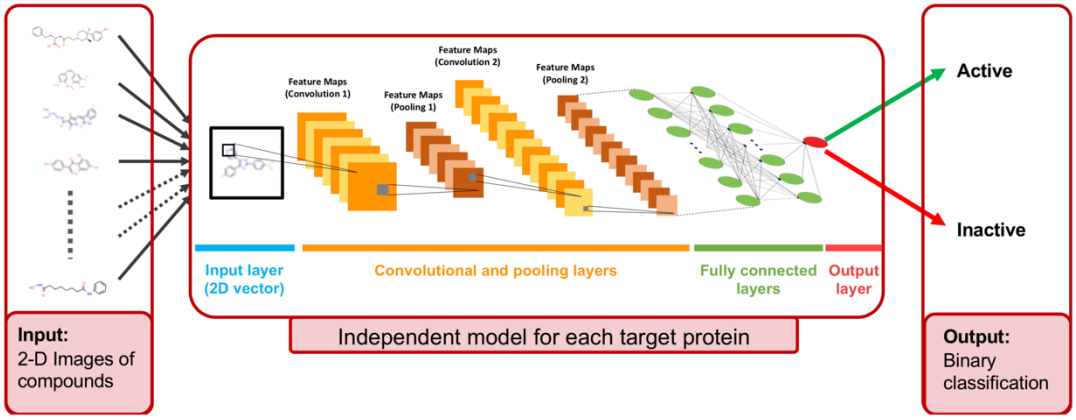


Figure 4.2. Illustration of the deep convolutional neural network structure of DEEPScreen, where the sole input is the 2-D structural images of the drugs and drug candidate compounds (generated from the SMILES representations as a data pre-processing step). Each target protein has an individual prediction model with specifically optimized hyper-parameters (please refer to the Methods section). For each query compound, the model produces a binary output either as active or inactive, considering the interaction with the corresponding target.

4.3.4 System Training and Test Procedures

For each target protein model, 80% of the training samples (from both the positives and the negatives datasets) were randomly selected as the training/validation dataset, and the remaining 20% was reserved for later use in the independent performance test procedure. Also, 80% of the training/validation dataset was employed for system training and 20% of this dataset was used for validation, during which the hyper-parameters of the models were optimized.

With the purpose of selecting the architecture(s) to be used in DEEPScreen, we initially trained and tested models for a small number of target proteins using a wide-range of hyper-parameters. At the end of these initial tests, we eliminated the AlexNET architecture since its performance was inferior to the performances of other two architectures. After this point, we continued our tests with Inception and our in-house DCNN architecture. We created and trained one model for each hyper-

parameter selection, for each target, for each architecture. The list of the hyper-parameters and the value selections are given in Table 4.2. The models were run on the validation datasets during training to obtain the predictive performance (i.e., accuracy, precision, recall, F1-score and MCC), which indicates the effectiveness of the pre-selected hyper-parameter values. At the end of the validation procedure, the best performing model (in terms of MCC) was selected for each target. At the end of this analysis, our in-house DCNN architecture was selected for 397 of the target proteins, and the Inception architecture was selected for the remaining 307 target proteins (out of the total of 704 targets). As a result, the finalized DEEPScreen system is composed of both Inception and in-house designed DCNN architectures. Next, test performances were calculated by running the finalized models on their corresponding independent test datasets, which have never been used before this point (i.e., performances reported in the Results Section). All of the training, test and prediction runs described in this study were carried out in parallel at the EMBL-EBI large-scale CPU cluster.

Table 4.2 Hyper-parameter types and the tested values during the training of DEEPScreen.

Hyper-parameter Name	**Test values**
Input Normalization	Yes No
Learning rate	0.0005 0.0001 0.005 0.001 0.01

Table 4.2 (continued)

Filter size	3 5
Stride	1
Padding	"same"
Number of convolutional layers	*
Number of filters in each convolutional layer	**
Number of neurons in each fully-connected layer	***
Optimizer	Adam (default) Momentum (default) RMSprop (default)
Mini-batch size	32 64
Drop-out rate	0.5 0.6 0.8
Batch Normalization	Yes

* Values between 3 and 8 were tested for the in-house DCNN architecture. For AlexNET and Inception, the default architectures were directly used without any change.

** Numerous "# of filter" value combinations were tested between 16 and 256.

*** For the two fully-connected layers (just before the output layer) in AlexNET, the number of neurons that were tested: (128,16), (256,128), (512,32), (1024,32) and (2048,2048). For the in-house DCNN, there were one fully connected-layer (before the output layer) and # of neurons tested were: 16, 32, 128, 256 and 512.

In order to investigate the possible reasons behind the performance differences between the Inception and the in-house DCNN architectures in DEEPScreen, we conducted a target protein family based comparison over our pre-trained 704 target protein models to observe if there is a performance difference between the two architectures for a specific protein family (i.e., for how many members of a target protein family the Inception model was the best performer, and for how many of them the in-house DCNN was the best). We found out that the architectures performed nearly the same for nuclear receptors. Considering the rest of the families, DCNN architecture performed better between 28% and 50%, compared to the Inception models. We believe the only reason behind observing this performance difference is that the Inception architecture is significantly more complex and computationally more demanding compared to the in-house DCNN architecture, as a result, the hyper-parameters space we were able to scan during the grid search analysis was smaller for Inception. A grid search with the same hyper-parameters space size for Inception models would probably result in predictive performances greater than or equal to the performance of the DCNN models. However, a grid search of this magnitude would require a very long time to finish even on a strong computing resource. To test this idea, we analyzed the Inception and in-house DCNN performances over 114 target proteins, all of which were difficult to model, as pointed out by the low predictive performances in our initial tests. For these 114 targets, we trained our predictive models and searched large hyper-parameter spaces for both the Inception and for the in-house DCNN models, and selected the best Inception and the best in-house DCNN for each of the 114 targets by checking the highest test performance in terms of the MCC measure. After that, we compared the best Inception model with the best in-house DCNN model, for each target (i.e., 114 comparisons in total). We found that in-house DCNN models performed better for

42 of the targets and the Inception model performed better for 35 of them (the performance was exactly the same for the remaining 37 targets). We also calculated the actual performance differences between the best in-house DCNN and the best Inception models for each target, and found that, the average performance difference was the same when we compared two groups: 1) targets, on which DCNN performed better, and 2) targets, on which Inception performed better. These results indicated that there is no significant performance difference between Inception and the in-house DCNN, when similar hyper-parameters spaces are searched during the model optimization step. The results of the Inception vs. in-house DCNN performance analysis has been provided in the repository of the study.

4.3.5 Benchmark Datasets for The Predictive Performance Comparison

All of the four non-random split datasets used in the performance analyses are constructed by considering scaffold/structure/temporal train-test sample divisions, as a result, they accurately simulate real-case prediction scenarios, where the predictive systems are queried with completely new compounds with different features (e.g., never-seen-before scaffolds).

First of all, we aimed to generate our own bias free benchmark dataset using our fundamental ChEMBL training set. For this, we first focused on further eliminating the negative selection bias, even though we previously showed that similarity among negative samples was around the same level as the similarity between negative (inactive) samples, in our fundamental datasets (please see the Results section), mainly due to the fact that we only included compounds with real experimental bioactivity measurements (coming from binding assays) against the intended target. For further elimination of negative selection bias, we identified the negative dataset compounds, whose all activity data points (against all targets) in the ChEMBL database are in the inactives range (i.e., ≥ 20 μM xC50) and discarded them. The compounds, which have at least one data point in the actives range (for any target) were kept in the negative datasets. Considering the rigorous filtering operations

applied to generate our source/fundamental bioactivity dataset (explained in the Methods section in detail), we assumed that even one active data point (i.e., ≤ 10 μM xC50) would be sufficient to accept that the corresponding molecule does not possess features that make it an all-inactive / invalid compound. To eliminate chemical bias from our datasets, we applied the Murcko scaffold [258] detection and train-test split (based on the detected scaffolds) module in the RDKit package. This way, for each target, all compounds with a distinct scaffold either ended up in the training set or in the test set; in other words, the compounds with the same scaffold were not distributed to both training and test. Following these rules, we carefully constructed train and test datasets for 17 representative targets spanning the main target families of enzymes, GPCRs, ion channels, nuclear receptors and others, with dataset sizes ranging from 143 to 5229. The total number of data points in the finalized dataset was 21200. The targets were selected mostly based on the representative drug targets list given in another study [259]. We selected 10 targets from the list given in Mysinger *et al.* (many of the remaining targets listed in this article were not among 704 DEEPScreen targets so they could not be covered), we additionally included Renin and JAK1 (since these two targets were also selected as use cases for further validation) and 5 additional randomly selected targets proteins (from different families), to reflect the target protein family distribution for 704 DEEPScreen targets. The gene names of the selected 17 targets are: MAPK14, JAK1, REN, DPP4, LTA4H, CYP3A4, CAMK2D, ADORA2A, ADRB1, NPY2R, CXCR4, KCNA5, GRIK1, ESR1, RARB, XIAP, NET; summing into 7 enzymes (taking the distribution of the enzyme sub-families into account as well), 4 GPCRs, 2 ion channels, 2 nuclear receptors and 2 others. We named this set as the representative targets benchmark dataset.

The second benchmark dataset we used in our study was directly obtained from the study by Lenselink *et al.* [231]. In this study, the authors created a high quality ChEMBL (v20) bioactivity dataset that includes 314,767 bioactivity measurements corresponding to target proteins with at least 30 bioactivity data points. They used pChEMBL = 6.5 (roughly 300 nM) bioactivity value threshold to create active and

inactive compound datasets for each target. The authors evaluated their method with a test dataset created by a temporal split, where for each target protein, all of the bioactivity data points reported in the literature prior to 2013 were used in the training, and the newer data points were gathered for the test dataset. This test dataset is more challenging for ML classifiers compared to any random-split dataset.

The third dataset we used was Maximum Unbiased Validation (MUV), another widely-used benchmark set, composed of active and inactive (decoy) compounds for 17 targets [260]. MUV dataset was generated from the PubChem Bioassay database. The active compounds in this dataset was selected to be structurally different from each other. Therefore, it is a challenging benchmark dataset, which avoids the bias rooting from highly similar compounds ending up in both training and test splits (i.e., chemical bias). There are 17 targets in MUV dataset, together with 30 actives and 15000 decoys for each target.

The fourth benchmarking dataset employed in this study was DUD-E, a well-known set for DTI prediction, which includes curated active and inactive compounds for 102 targets. The active compounds for each target was selected by first clustering all active compounds based on the scaffold similarity and selecting representative actives from each cluster. The inactive compounds were selected to be similar to the active compounds in terms of the physicochemical descriptors, but dissimilar considering the 2-D fingerprints [259]. The benchmark dataset consists of 102 targets, 22,886 actives (an average of 224 actives per target) and 50 property-matched decoys for each active, which were obtained from the ZINC database [259]. It is also important to note that DUD-E benchmark dataset is reported to suffer from negative selection bias problem, as a result, we did not conclude our results on the performance on the DUD-E dataset. We just used DUD-E dataset to make a highly generic performance comparison with the literature, since DUD-E is a widely used benchmark dataset.

4.3.6 Literature Based Validation of Novel DTI Predictions

DEEPScreen produced 21.2 million completely novel DTI predictions. As a result, it was not possible to manually check the literature if a research group has already studied these specific drug/compound-target interactions for validation. Instead we assumed a more directed approach, where the validation cases were determined from a newer version of ChEMBL and from the literature first, then, DEEPScreen's novel prediction results were searched to observe if these interactions were identified by DEEPScreen as well. The selected cases are composed of two types of data points. The first one concerns the already approved drugs (or the ones in the experimental development phases), where the given target interactions are novel (i.e., not part of the already approved or experimental treatment for these drugs), thus, serve the purposes of drug repositioning. For this, we found the cases where the corresponding drug has bioactivity data points for new targets in ChEMBL v24, that was not part of v23 (ChEMBL v23 was used for the training of DEEPScreen). As such, these cases correspond to the recently curated data. Using this set, we only selected the cases where the corresponding targets were among the 704 target proteins of DEEPScreen, and the source publications of the reported bioactivities were novel (i.e., from 2016 and 2017). It was not possible to find any cases with 2018 publications since these articles are not curated in ChEMBL yet. We then searched DEEPScreen large-scale prediction results to find if these cases were predicted. The results only display a few of the coinciding data points with the most novel source publications. The second type of data points consist of completely novel bio-interactions that has not entered ChEMBL or any other bioactivity database yet. Since these compounds are not incorporated in ChEMBL, our large-scale prediction results did not include them. To observe if DEEPScreen can predict the reported activities given in 2 selected drug design and development publications from 2018 [261], [262], we generated the SMILES representations and the 2-D structural images of the documented compounds using their molecular formula as reported in the corresponding publications. After that, we run the query compounds against their

newly identified targets (which were reported in the respective articles) to see if DEEPScreen can predict these highly novel interactions. For the literature-based validation analysis, the approved and experimental drug information was obtained from the DrugBank database [141].

4.3.7 Performance Evaluation Metrics

We mainly used 3 evaluation metrics, F1-score, Matthews Correlation Coefficient (MCC) and area under receiver operating characteristic curve (AUROC) to evaluate the predictive performance of DEEPScreen and to compare its results with other DTI prediction methods. The formulas of these evaluation metrics are given below together with precision and recall that make up F1-score:

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

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

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

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

In the equations above, TP (i.e., true positive) represents the number of correctly predicted interacting drug/compound-target pairs, FN (i.e., false negative) represents the number of interacting drug/compound-target pairs, that are predicted as non-interacting (i.e., inactive). TN (i.e., true negative) denotes the number of correctly predicted non-interacting drug/compound-target pairs, whereas FP (i.e., false positive) represents the number of non-interacting drug/compound-target pairs, which are predicted as interacting.

4.4 Results

4.4.1 Drug-Target Interaction Prediction with DEEPScreen

In this study, we approached DTI prediction as a binary classification problem. DEEPScreen is a collection of DCNNs, each of which is an individual predictor for a target protein. The system takes drugs or drug candidate compounds in the form of SMILES representations as query, generates 200-by-200 pixel 2-D structural/molecular images using SMILES, run the predictive DCNN models on the input 2-D images, and generates binary predictions as active (i.e., interacting) or inactive (i.e., non-interacting) for the corresponding target protein (Figure 4.2). In order to train the target specific predictive models of DEEPScreen with a reliable learning set, manually curated bio-interaction data points were obtained from the ChEMBL bioactivity database, and extensively filtered (Figure 4.1). The technical details regarding both the methodology and the data is given in the Methods section. Following the preparation of datasets, we extracted target protein based statistics, in terms of amino acid sequences [11], domains [263],[168], functions, interacting compounds and disease indications [264],[265]. The results of this analysis can be found the manuscript [266].

4.4.2 System Robustness Against Input Image Transformations

We also carried out several tests to examine the robustness of the DEEPScreen system against input image transformations, since this is a critical topic for CNN architectures that process 2-D images. One of the critical points in the computer vision tasks is the system robustness concerning the differences in the representations of the object of interest, such as the viewing angle or the scale. In DEEPScreen, input images are standardized by computationally generating them from SMILES representations, this way all images have similar representations in terms of viewing angle (i.e., rotation). However, we investigated the question of how

the models would behave if they are provided with rotated compound images as the query. For this, we have selected 3 target protein models (i.e., BCHE gene - CHEMBL5077, GSK3beta gene - CHEMBL3638364, PTGS1 gene - CHEMBL221) and we constructed the rotated compound images of the positive and negative performance test dataset compounds of these targets. 7 new samples were generated from each test compound image, by rotating the original image by 45 degree angle. We fed these rotated images to the original pre-trained predictive models as the query set. Since the original models have never seen these rotated images before (during training), the performances were decreased by 29%, 23% and 32% in terms of the MCC measure and by 22%, 18% and 18% in terms of accuracy for BCHE, GSK3beta and PTGS1 models, respectively (compared to the original test performances of these models, when there is no rotated images in the query set). It was argued in the literature that the application of training data augmentation by generating and adding new samples to the training set by rotating the existing images solves this problem [267]. To observe if this is the case for DEEPScreen, we generated rotated compound images for each and every positive and negative training dataset instance (using the same 45 degree rotation approach), and re-trained the same predictive models using the enhanced training datasets and performed the hyper-parameter optimization tests with grid-search. After that, we measured the performance of these newly trained models by querying them with the rotation-added test datasets used in the previous test. Finally, we compared the performance of the rotation-trained models with the performance of the original models (i.e., models without any rotated images in training or in test datasets). The results showed that, when the rotated images were added to the training, the performance remained roughly the same for all 3 selected models (i.e., 0-2% performance decrease in both MCC and accuracy for BCHE, GSK3beta and PTGS1), which indicates that training with data augmentation by generating rotated data points worked well. However, it was not possible to apply this methodology to train all of 704 target protein models of DEEPScreen due to significantly increased computational complexity. The application of rotated compound images increased the training dataset size of each model 8 times relative

to the original training datasets. This was a huge burden especially for the complex Inception architecture-based models. Considering the fact that whole hyper-parameter optimization by grid search procedure should have been repeated, it was not possible to construct a rotation invariant system for DEEPScreen in the end. Instead, we relied on generating canonical images for both the training and query compounds from SMILES as the input, which worked well in practice. There are also novel alternative solutions proposed lately in the literature such as the approach proposed by Thomas *et al.*, where the authors developed a DCNN architecture that is equivariant to rotations by using filters from spherical harmonics [268]. Similar approaches may be applied in the future to modify DEEPScreen to make it completely rotation invariant so that the user drawn images can directly be fed to the system as the input (through a web-interface) instead of SMILES.

Another important transformation type for computer vision tasks is the scaling. In DEEPScreen, the compounds are drawn as images by fully occupying the 200-by-200 pixel frame no matter what their actual molecular sizes are. This means that a certain component (i.e., a sub-structure such as a benzene ring) in a large (i.e., high molecular weight) compound will occupy fewer number of pixels (i.e., appear smaller in the image) compared to the size of the same sub-structure as a part of a smaller (i.e., low molecular weight) compound. The predictive system should be invariant to these scaling variances, in other words, it should perceive, for example, a benzene ring structure independent from its size on the compound images. This is generally achieved by training CNN-based systems by using input instances containing the features of interest in different scales [267]. In DEEPScreen, this is automatically achieved since the training dataset compounds of each target protein contain both relatively larger and smaller molecules. In order to examine this issue, we conducted a scaling analysis on our case study model: renin. In this analysis, we augmented the test dataset of renin target protein model by scaling both the positive and negative test dataset molecules by decreasing the sizes by 10%, 20% and 30% (i.e., molecules occupy a smaller area on the 200-by-200 pixel images). In the end, we obtained a test dataset 4 times larger compared to the original set (i.e., original +

10% + 20% + 30% scaled down compounds for all test samples). We fed the newly generated scaled test dataset as query to the original pre-trained predictive model of renin target protein. The assumption here was that, since the training dataset compounds of renin contained samples in different scales (i.e., same sub-structures automatically drawn in different sizes in different images due their presence in both large and small compounds) the system will be robust against the variance in the artificially scaled samples in the test dataset. The performance results of this test have indicated that decreasing the molecule size by 10% affected the predictive performance by reducing precision, recall, F1-score and accuracy by 9%, 7%, 8% and 9%, respectively. A same trend was observed for 20% and 30% size reductions (i.e., 8-10% performance reduction compared to the test results of the previous scaling), as well, pointing out a linear relation between molecule size scales and the performance. The results pointed out that the performance change observed with a 10% scaling is acceptable.

Nevertheless, it is possible to query a target protein model with a compound that is significantly larger or smaller compared to all training dataset samples of the model. In this case, the system may misinterpret a feature/sub-structure since it will be scaled very differently compared to the scales it is aware of (from the training data). However, a molecule that is significantly larger or smaller compared to all known ligands of a target would be less likely to interact with the intended target due to its inability to occupy the intended binding region/pocket. In order to test this, we analyzed the size distribution of the compounds in the active and inactive training datasets of renin, in terms of molecular weights, since the molecular weight can be a good indicator of the scale of the compounds on the 2-D images. The molecular weight is only consist of a number (i.e., we can compare different molecular weights on a 1 dimensional space), whereas in images, compounds are represented over 2 dimensions (where there is more space), as a result, 2-D scaling differences are roughly equivalent to the square of the molecular weight differences (e.g., a 10% size difference in 2-D images is roughly equivalent to 31.5% difference in molecular weight). We also manually checked several compound images from the original

training datasets of different target proteins and observed that the abovementioned relation between the molecular weight change and 2-D scaling change holds true.

The analysis of the training dataset compounds of an example target protein: "Complement factor D" (gene name: CFD, ChEMBL id: CHEMBL2176771), which has 732 active ligands (close to the average number of active ligands for all DEEPScreen targets: 728), revealed that the molecular weight distribution is normal and the active compounds have a mean of 504 g/mol and a standard deviation of 38 g/mol, which indicates a weight change of 26% between +/- two standard deviations from the mean (i.e., 580 to 428 g/mol), translating into roughly 7% (i.e., 0.26^2) scaling difference on 2-D. It is important to note that two standard deviations from the mean covers nearly 95% of all data points in a normal distribution. This places the expected scaling difference between the potential ligands of a target within an acceptable zone in terms of model performance reduction (i.e., < 10%).

4.4.3 Sources of Dataset Bias in Model Evaluation

Labelled ground-truth data are split into training/validation/test partitions in order to train, optimize and evaluate predictive models. There are two basic strategies in the field of virtual screening (or DTI prediction) in terms of dataset split. The first and the most basic one is the random-split, where the data points are separated randomly without any particular consideration. Evaluations using random-split datasets are good indicators of what would be the model performance in predicting new binders that are structurally similar (e.g., containing the same scaffolds) to the compounds in the training dataset. The second widely used data split strategy in DTI prediction is the similarity-based (or non-random) split, where data points are divided according to similarities between compounds/targets/bioactivities, according to the assumed modelling approach. Here, the aim is to prevent very similar data points from ending up both in training and test sets. In ligand-based prediction approaches (such as DEEPScreen), the input samples are compounds, as a result, datasets are split according to molecular similarities between compounds. This can be done by

checking the shared scaffolds in these compounds and applying a scaffold-based split or by calculating pairwise structural similarities and clustering the compounds based on this.

There are critical points and risks in constructing training and test datasets for developing a virtual screening system and analysing its predictive performance. The first risk would be the introduction of chemical bias to the tests, where structurally similar compounds end up both in training and test datasets. This often makes the task of accurate prediction a somewhat trivial task, since structurally similar compounds usually have similar (or the same) targets. Random-split datasets usually suffer from this problem. Another risk is the negative selection bias, where negative samples (i.e., inactive or non-binder compounds) in the training and/or test datasets are structurally similar to each other in a way, which is completely unrelated to their binding related properties [269]. So that, a machine learning classifier can easily exploit this feature to successfully separate them from the positives. Both of these cases would result an overestimation of the model performance during benchmarks, especially when the tests are made to infer to performance of the models in predicting completely novel binders to the modelled target proteins. It was reported that a widely used benchmark dataset DUD-E [259] suffers from the negative selection bias problem, even though the chemical bias issue was properly addressed during the construction of this benchmark. In DUD-E, most of the property matched decoys (i.e., negatives) were found to be highly biased, as the models trained on specific targets were highly successful in identifying the negatives of completely different targets [269]. In other words, most of the decoys shared features that make them non-binders to nearly all target proteins, and care should be taken while evaluating predictive models on this benchmark. In this study, we evaluated the performance of DEEPScreen on 5 different datasets (e.g., large-scale random-split dataset, both chemical and negative selection bias free representative targets dataset, ChEMBL temporal/time split dataset, MUV and DUD-E) in order to observe the behaviour of the system and its comparison with the state-of-the-art on benchmarks with differing

strengths and weaknesses. The content and properties of these datasets are explained in the Methods section.

4.4.4 Analysis of the DEEPScreen Dataset in terms of Negative Selection Bias

To examine DEEPScreen source dataset in terms of negative selection bias, we compared the average molecular similarities among the member compounds of each target specific negative training dataset, also, we make a cross comparison of average molecular similarity of the compounds in the positive training dataset a target against the compounds in the negative training dataset of the same target, to uncover if there is a statistically significant structural difference between positives and negatives. For this, we employed Morgan Fingerprints (ECFP4) and the pairwise Tanimoto similarity calculation between all compound pair combinations. According to the results of this analyses on the datasets of 704 target proteins, there was no target where the inactive training dataset compounds are more similar to each other compared to the inter group similarities between the active and inactive dataset compounds of that target protein model, with statistical significance according to t-test (at 95% confidence interval). Actually, mean active to inactive similarity was higher than the similarity among the inactives for 211 targets, indicating that inactives do not share a global similarity that separates them from actives, that would otherwise make it easy to distinguish them, and introduce a bias to the performance analysis. These results are displayed in Figure 4.3 as target based mean pairwise compound similarity curves for intra-group (among inactives) and inter-group (actives to inactives) similarities with error bands. The most probable reason behind the observation of no significant difference was that, we directly used the experimental bioassay results reported in the ChEMBL database to construct our negative datasets by setting an activity threshold (i.e., ≤ 10 μM), instead of manually constructing decoy datasets. Thus, the compounds in our negative datasets are able

to interact with the intended targets, with very low affinities. The results indicated that the negative selection bias is not an issue for the DEEPScreen source dataset.

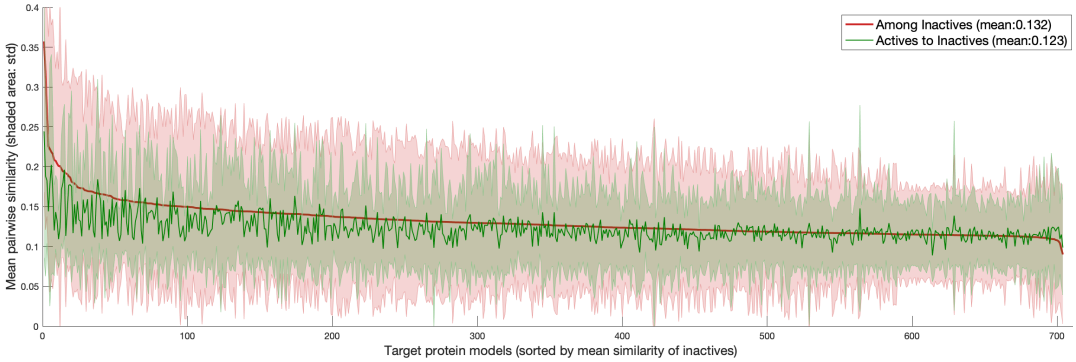


Figure 4.3. Target-based (x-axis) average pairwise compound similarity (y-axis) curves for intra-group (among inactives) and inter-group (actives to inactives) similarities with error bands for 704 DEEPScreen targets.

4.4.5 Performance Evaluation of DEEPScreen and Comparison with Other Methods

4.4.5.1 Large-scale Performance Evaluation and Comparison on Random-Split Dataset

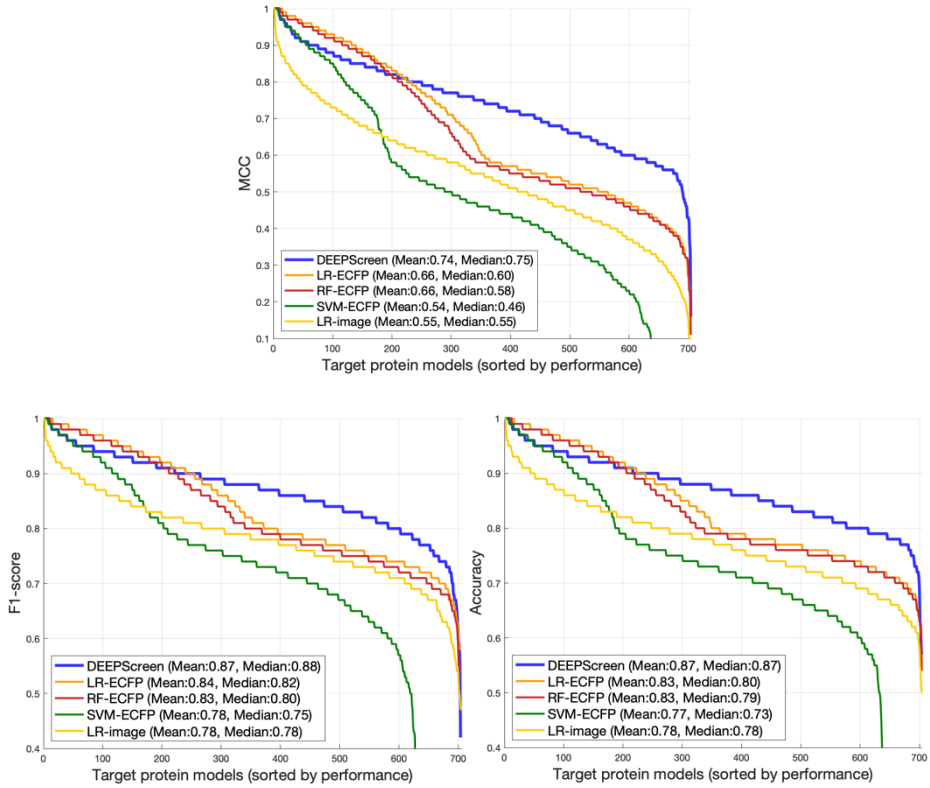
According to our basic performance tests, for 613 of the target protein models (out of 704), DEEPScreen scored an accuracy ≥ 0.8, with an overall average accuracy: 0.87, F1-score: 0.87 and Matthews Correlation Coefficient (MCC): 0.74. Additionally, high-level target protein family based average model performances indicated that DEEPScreen performs sufficiently well on all target families (average MCC for enzymes: 0.71, GPCR: 0.80, ion channels: 0.76, nuclear receptors: 0.76, others: 0.69). All performance evaluation metrics used in this study are explained in the Methods section.

Following the calculation of DEEPScreen's performance, we compared it against conventional DTI prediction approaches (classifiers: random forest - RF, support

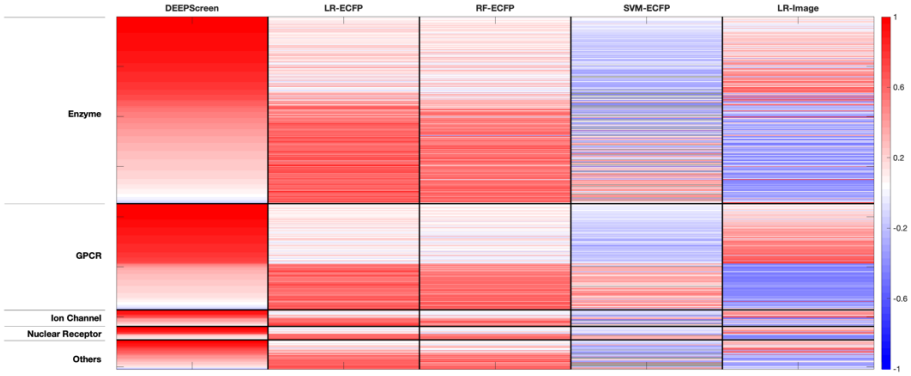
vector machines - SVM and logistic regression - LR) using the exact same random-split training/test sets under two different settings. In the first setting, conventional classifiers were trained with circular fingerprints (i.e., ECFP4 [247]) of the compounds, which represents the current state-of-the-art in DTI prediction. The model parameters of the conventional classifiers were optimized on the validation dataset and the finalized performances were measured using the independent test dataset, similar to the evaluation of DEEPScreen. In the second setting, the same feature type (i.e., 2-D molecular representations) is employed. These conventional classifiers normally accept 1-D (column-type) feature vectors, therefore, we flattened our 200-by-200 images to be used as input. Thus, the performance comparison solely reflects the gain of employing DCNNs as opposed to conventional/shallow classification techniques. It is possible to argue that conventional classifiers such as LR, RF and SVM may not directly learn the from the raw image features, and thus, sophisticated image pre-processing applications, such as constructing and using histogram of oriented gradients [270], are required to train proper image feature based predictive models. Here, our aim was to identify the most prominent factor behind the performance increase yielded by DEEPScreen (i.e., is it only the use of DNNs, mostly independent from the featurization approach, or is it the use of image-based features together with the employment of DNNs to classify them), without a possible affect from a third-party data processing application. As a result, we directly used the raw image features. Figure 4.4.a displays the overall ranked target based predictive performance curves, in MCC, accuracy and F1-score, respectively. We did not include RF-Image and SVM-Image performance in Figure 4.4 since RF models performed very similar to the LR models on nearly all models, and SVM models were unable to learn the hidden features in most of the cases and provided a very low performance. It is possible to observe the results of RF-Image and SVM-Image in the performance tables provided in the repository of this study. DEEPScreen performed better compared to all conventional classifiers employed in the test according to both mean and median performance measures. Especially, the performance difference was significant when MCC was

used, which is considered to be a good descriptor of DTI prediction performance. For all performance measures, among the best 200 target models for each method, LR-ECFP and RF-ECFP models have higher performance compared to DEEPScreen, however, DEEPScreen takes over after the 200th model and displayed a much better performance afterwards. Overall, DEEPScreen performed 12% and 23% better in terms of mean and median performances respectively, compared to its closest competitors (i.e., LR-ECFP and RF-ECFP) in terms of MCC. According to our results, the best classifier was DEEPScreen for 356 targets (LR-ECFP for 250, RF-ECFP for 141, SVM-ECFP for 24 targets). The results indicate that DEEPScreen's performance is stable over the whole target set. On the other hand, state-of-the-art classifiers perform very well for some targets but quite bad at others, pointing out the issues related to generalization of conventional fingerprints.

(a)



(b)



(c)

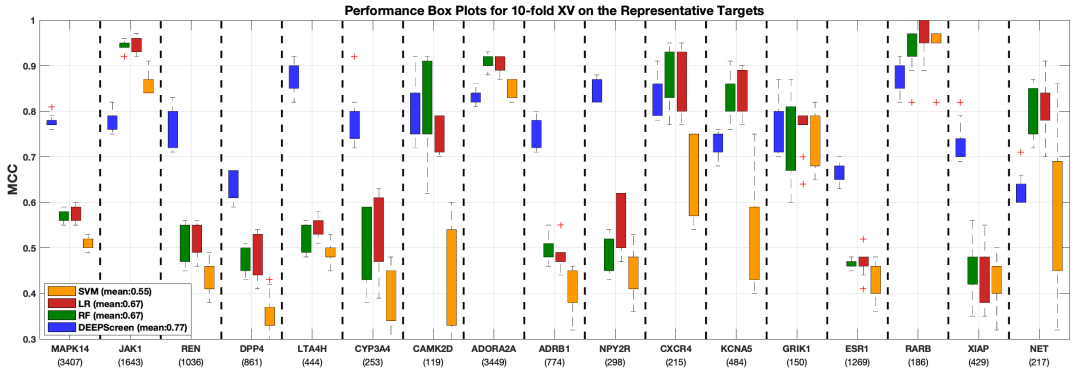


Figure 4.4. (a) DEEPScreen vs. state-of-the-art classifiers overall predictive performance comparison. Each point in the horizontal axis represents a target protein model, the vertical axis represents performance in: MCC, accuracy and F1-score, respectively. For each classifier, targets are ranked in a descending performance order. Average performance values (mean and median) are given inside the plots. (b) Target-based maximum predictive performance (MCC-based) heatmap for DEEPScreen and conventional classifiers (columns) (LR: logistic regression, RF: random forest, SVM: support vector machine; ECFP: fingerprint-based models, Image: 2-D structural representation-based models). For each target protein (row), classifier performances are shown in the shades of red (i.e., high performance) and blue (i.e., low performance) colours according to Z-scores (Z-scores are calculated individually for each target). Rows are arranged in blocks according to target

families. The height of a block is proportional to the number of targets in its corresponding family (enzymes: 374, GPCRs: 212, ion channels: 33, nuclear receptors: 27, others: 58). Within each block, targets are arranged according to descending performance from top down with respect to DEEPScreen. Grey colour signifies the cases, where learning was not possible. (c) MCC performance box plots in 10-fold cross-validation experiment, to compare DEEPScreen with the state-of-the-art DTI predictors.

Figure 4.4.b shows target protein based predictive performance (in terms of MCC) z-score heatmap for DEEPScreen and the conventional classifiers, where each horizontal block corresponds to a target family. As displayed in Figure 4.4.b, DEEPScreen performed significantly better for all families (solid red blocks), LR-ECFP and RF-ECFP came second, LR-Image took the third place, and SVM-ECFP came at the last place. An interesting observation here is that, image-based (i.e., DEEPScreen and LR-Image) and fingerprint-based classifiers display opposite trends in predictive performance for all families, indicating that the image-based approach complements the fingerprint approach. Also, LR-ECFP and LR-Image performances were mostly opposite, indicating a pronounced difference between the information obtained from fingerprints and images. Although LR-Image's overall performance was lower compared to LR-ECFP, it was still higher compared to SVM-ECFP, implying LR-Image managed to learn at least part of the relevant hidden features. There was no significant difference between the protein families in terms of the classifier rankings, though, DEEPScreen's domination was slightly more pronounced on the families of GPCR, ion channel, and nuclear receptor.

In order to compare the performance of DEEPScreen with the conventional classifiers on a statistical basis, we carried out 10 fold cross-validation on the fundamental random-split datasets of the same 17 representative target proteins (i.e., gene names: MAPK14, JAK1, REN, DPP4, LTA4H, CYP3A4, CAMK2D, ADORA2A, ADRB1, NPY2R, CXCR4, KCNA5, GRIK1, ESR1, RARB, XIAP, NET) that were employed for the construction of chemical and negative selection bias free scaffold-split benchmark dataset (please see Methods section for

information about the selection procedure for these target proteins). We applied Bonferroni corrected t-tests to compare the performance distribution of each method on each target independently (10 measurements from each 10-fold cross-validation experiment constitute a distribution). The statistical tests were conducted on MCC performance metric due to its stability under varying dataset size partitions. Figure 4.4.c displays the MCC performance results as box plots, for 17 targets. Each box represents a classifier's 10 MCC measures on 10 different folds of a target's training dataset, in the cross-validation. In these plots, the top and bottom borders of the box indicate the 75^{th} and 25^{th} percentiles, the whiskers shows the extension of the most extreme data points that are not outliers, and plus symbols indicate outliers. The number written under the gene names of the respective targets indicate the size of the training datasets (actives). According to results, there was no observable relation between dataset sizes and a classifier's performance. According to the results of the multiple pairwise comparison test (Bonferroni corrected t-tests), DEEPScreen performed significantly better (compared to the best conventional classifier for each target) for 9 of the 17 representative targets (i.e., genes: MAPK14, REN, DPP4, LTA4H, CYP3A4, ADRB1, NPY2R, ESR1, XIAP), which constitutes 71%, 50%, 50% and 50% of enzymes, GPCRs, nuclear receptors and 'others' families, respectively (p-value < 0.001). Whereas, the best conventional classifier managed to significantly beat DEEPScreen only on 2 representative targets (i.e., genes: JAK1 and RARB), which constitute 14% and 25% of enzymes and GPCRs, respectively (p-value < 0.001). For the rest of the representatives (6 targets), there was no statistically significant difference between DEEPScreen and the conventional classifiers. The results indicate that DEEPScreen's dominance is mostly statistically significant.

To examine the test results in relation to potential performance effecting factors, we first checked the correlation between the performances of different classifiers to observe the overlap and the complementarity between different ML algorithms and featurization approaches. Spearman rank correlation between the performance (MCC) distribution of DEEPScreen and the state-of-the-art (i.e., LR, RF and SVM

with fingerprint-based features) were around 0.25 (against LR-ECFP and RF-ECFP) and 0.51 (against SVM-ECFP), indicating only a slight relation, and thus, a potential complementarity (as also indicated in Figure 4.4.b). However, the rank correlation between LR-ECFP and RF-ECFP was 0.97 indicating a high amount of overlap and possibly no complementarity. The correlation between LR-ECFP (or RF-ECFP) and SVM-ECFP was around 0.62, just slightly higher than DEEPScreen vs. SVM-ECFP. It was interesting to observe DEEPScreen's performance rank was more similar to SVM-ECFP than LR-ECFP or RF-ECFP. To check if the difference between DEEPScreen and LR/RF is due to the employed algorithmic approach or due to the featurization approach, we checked the correlation between DEEPScreen and LR that used image features (i.e., LR-Image), which resulted in a correlation value of 0.68. Whereas, the rank correlation between LR-ECFP and LR-Image was only 0.21. These results demonstrated that the low correlation between DEEPScreen and LR-ECFP (or RF-ECFP) was mainly due to the difference in featurization, and there is possibly a complementarity between the featurization approaches of using molecular structure fingerprints and 2-D images of compounds. Also, the observed high performance of DEEPScreen indicated that deep convolutional neural networks are successful in extracting knowledge directly from the 2-D compound images.

Subsequently, we checked if there is a relation between training dataset sizes and the performance of the models, since deep learning-based methods are often reported to work well with large training sets. For this, we calculated the Spearman rank correlation between DEEPScreen performance (MCC) and the dataset sizes of 704 target proteins, and the resulting value was -0.02, indicating no correlation. The results were similar when LR and RF were tested against the dataset sizes (-0.08 and -0.02, respectively). However, the result for SVM was 0.20, indicating a slight correlation. Finally, we checked the average dataset size of 356 target proteins, on which DEEPScreen performed better (MCC) compared to all conventional classifiers and found the mean value as 629 active compounds, we also calculated the average dataset size of the models where the state-of-the-art approaches performed better compared to DEEPScreen, and found the mean value as 542 active

compounds. The difference in the mean dataset sizes indicates DEEPScreen performs generally better on larger datasets.

Next, we applied a statistical test to observe if there are significantly enriched compound scaffolds in the training datasets of target proteins, where DEEPScreen performed better compared to the state-of-the-art approaches. For this, we first extracted Murcko scaffolds [258] of both active and inactive compounds of 704 DEEPScreen targets, using the RDkit scaffold module. Scaffold extraction resulted in o total of 114269 unique Murcko scaffolds for 294191 compounds. Then, we divided each scaffold's statistics to four groups: *(i)* the number of occurrences in the active compound datasets of targets where DEEPScreen performed better, *(ii)* the number of occurrences in the active compound datasets of targets where the state-of-the-art classifiers performed better, *(iii)* the number of occurrences in the inactive compound datasets of targets where DEEPScreen performed better, and *(iv)* the number of occurrences in the inactive compound datasets of targets where state-of-the-art classifiers performed better. Using these four groups, we calculated the Fisher's exact test significance (p-value) for the decision on the null hypothesis that there are no non-random associations between the occurrence of the corresponding scaffold in the DEEPScreen dominated target models and the state-of-the-art classifier dominated models. With a p-value threshold of 1x10-5, we identified 140 scaffolds, 61 of which were enriched in the DEEPScreen dominated target models. With the aim of reducing the extremely high number of unique scaffolds, we repeated the exact same procedure by using the generalized versions of the identified scaffolds. The generalization procedure (using RDkit) reduced the number of unique scaffolds to 55813. The statistical test resulted in a total of 211 significant generalized scaffolds, 101 of which were enriched in the DEEPScreen dominated target models. Although we managed to identify several significant scaffolds, most of them were presented in the datasets of only a few targets. The most probable reason behind was the high diversity of compounds in the DEEPScreen training datasets. SMILES representations of significant scaffolds and significant generalized

scaffolds are given together with their respective p-values in tabular format, in the repository of DEEPScreen.

4.4.5.2 Performance Evaluation and Comparison on Similarity-based Split Datasets

We compared the results of DEEPScreen with multiple state-of-the-art methods and highly novel DL-based DTI prediction approaches by employing four non-random split datasets (i.e., representative targets benchmark, temporal/time split dataset, MUV and DUD-E).

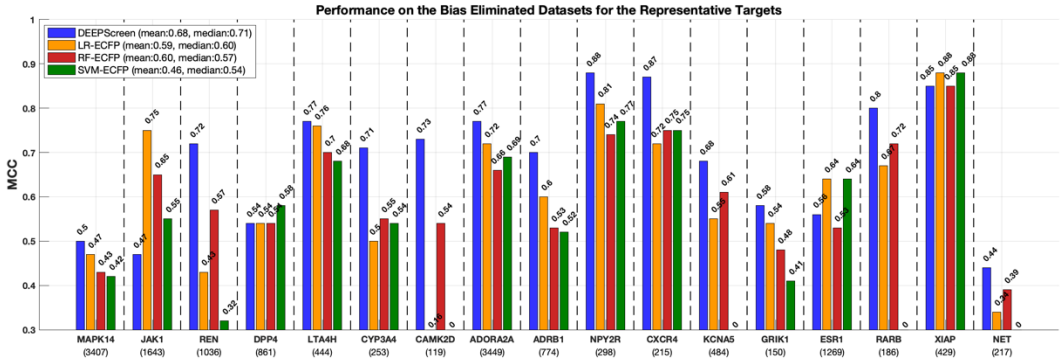
4.4.5.3 Comparison with the State-of-the-art Using our Scaffold Split Dataset

In order to test DEEPScreen free from chemical and negative selection biases, and to identify its potential to predict completely novel interacting drug candidate compounds for the intended target proteins, we carefully constructed target specific active/inactive compound datasets with a structural train-test split and collectively named it the representative targets benchmark dataset (please see the Methods section for more information on this dataset). The newly constructed representative targets benchmark dataset was used to train and test DEEPScreen along with the same state-of-the-art approaches used in virtual screening (i.e., LR, RF and SVM with fingerprint-based features). Figure 4.5.a displays the performance results (MCC) on different representative targets. As observed, on average, DEEPScreen was the best performer with a median MCC of 0.71, whereas, the best state-of-the-art method LR scored a median MCC of 0.6. RF performed similar to LR on average and on most of the targets individually, and SVM could not manage to learn from the challenging datasets of 4 targets, where it scored MCC = 0. Out of the 17 representative targets, DEEPScreen was the best performer for 13 of them, where the combined performance of the state-of-the-art methods managed to beat DEEPScreen

on 4 targets. Considering the target protein families, DEEPScreen was the best performer for 71% of the enzymes, 100% of GPCRs and ion channels, 50% of the nuclear receptors and 'others' families. The results indicate the effectiveness of the proposed approach in terms of producing interacting compound predictions with completely different scaffolds compared to the scaffolds present in the training datasets. Chemical and negative bias eliminated representative targets benchmark datasets is shared in the repository of DEEPScreen.

To benchmark DEEPScreen on an additional structural train-test split dataset, and to compare it with the state-of-the-art, we employed the Maximum Unbiased Validation (MUV) dataset. Since MUV is a standard reference dataset that is frequently used to test virtual screening methods, our results are also comparable with other works that employed the MUV benchmark. We trained DEEPScreen prediction models for 17 MUV targets using the given training split and calculated performance on the test split. We repeated the procedure using the conventional classifiers LR and RF that use fingerprint feature vectors. We left SVM out of this analysis based on its significantly inferior performance in the previous tests. The MUV performance results are shown in Figure 4.5.b with MCC bar plots for DEEPScreen, LR and RF. As observed from this figure, DEEPScreen had a higher performance on 15 out 17 targets, DEEPScreen and RF had the same performance on 1 target and there was a performance draw on the remaining target. Out of the 15 targets that DEEPScreen performed better, the performance difference was highly pronounced on 14 of them. The mean MCC for DEEPScreen, LR and RF were 0.81, 0.43 and 0.63, respectively; indicating a clear performance difference on a bias free benchmark dataset.

(a)



(b)

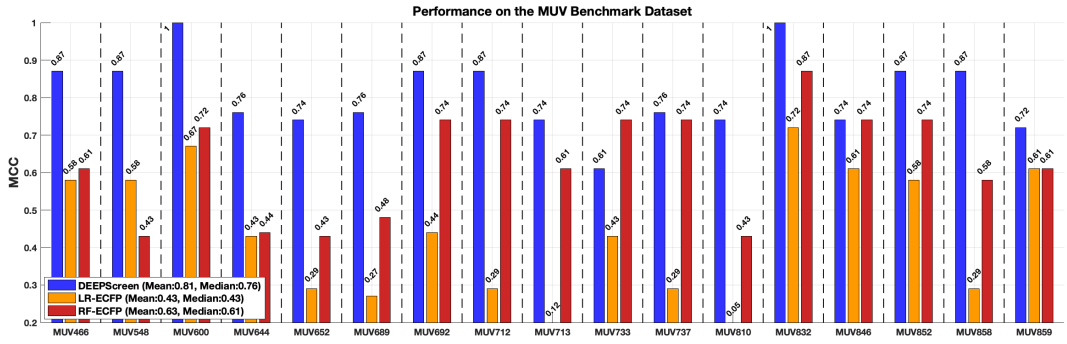


Figure 4.5. Predictive Performance evaluation and comparison of DEEPScreen against the state-of-the-art DTI prediction approaches, on scaffold-split benchmarks: (a) Bar plots of MCC values on representative targets dataset; (b) Bar plots of MCC values on the MUV dataset.

4.4.5.4 Comparison with Novel DL-based DTI Prediction Methods Using Multiple Benchmarks

For the DL-based DTI prediction method comparison analysis, we employed three benchmarks: temporal split, MUV and DUD-E (please refer to the Methods section for more information on these benchmark sets). We re-trained and tested DEEPScreen using the exact same experimental settings and evaluation metrics that were described in the respective articles [121], [123], [134], [231], [232]. Two of these datasets (i.e., MUV and DUD-E) are frequently employed in DTI prediction studies and the performance results of DEEPScreen on these datasets will also be comparable with future studies, where the same benchmark sets (together with the same train/test methodology) are employed. The results of this analysis reflect both the benefits of using 2-D images of compounds as the input, and the constructed DCNN-based architecture. It is important to mention that in each of these benchmark tests, DEEPScreen was trained with only the training portion of the corresponding benchmark dataset (i.e., MUV, DUD-E or ChEMBL temporal split set); in other words, our fundamental training dataset was not used at all. As a result, the number of training instances was significantly lower, which resulted in lower performances compared to what could have been achieved by using the regular predictive models of DEEPScreen.

Table 4.3 shows the results of DEEPScreen along with the performances reported in the respective articles (including both novel DL-based methods and the state-of-the-art approaches). As shown, DEEPScreen performed significantly better compared to all methods on the ChEMBL temporal split dataset. Lenselink *et al.* employed Morgan fingerprints (i.e., ECFPs [247]) at the input level as the compound feature, which currently is the most widely used (state-of-the-art) ligand feature type for DTI prediction. On their temporal split test dataset, DEEPScreen performed 36% better compared to the best model in the study by Lenselink *et al.* (i.e., multi-task DNN PCM – proteochemometics, also a deep learning based classifier), indicating the effectiveness of employing 2-D image-based representations as input features.

Table 4.3 The average predictive performance comparison between DEEPScreen and various novel DL-based and conventional DTI predictors.

Dataset	**Reference**	**Method/Architecture**	**Performance (Metric)**
ChEMBL Temporal-split Dataset		DEEPScreen: DCNN with 2-D Images	**0.45 (MCC)**
	Lenselink *et al.* [231]	Feed-forward DNN PCM (best model)	0.33 (MCC)
		Feed-forward DNN	0.30 (MCC)
		SVM	0.29 (MCC)
		LR	0.26 (MCC)
		RF	0.26 (MCC)
		Naïve Bayes	0.10 (MCC)
Maximum Unbiased Validation (MUV) Dataset		DEEPScreen: DCNN with 2-D Images	**0.88 (AUROC)**
	Kearnes *et al.* [134]	Graph convolution NNs (W_2N_2)	0.85 (AUROC)
	Ramsundar *et al.* [121]	Pyramidal Multitask Neural Net (PMTNN)	0.84 (AUROC)
		Multitask Neural Net	0.80 (AUROC)
		Single-Task Neural Net	0.73 (AUROC)
		RF	0.77 (AUROC)
		LR	0.75 (AUROC)

DEEPScreen was the best performer on the MUV dataset (Table 4.3), by a small margin, compared to the graph convolutional neural network (GCNN) architecture proposed by Kearnes *et al.* [134]. It is interesting to compare DEEPScreen with GCNN models since both methods directly utilize the ligand atoms and their bonding information at the input level, with different technical featurization strategies. Nevertheless, the classification performance of both methods on the MUV dataset was extremely high and more challenging benchmark datasets are required to analyse their differences comprehensively. The performance difference between DEEPScreen (or GCNN) and most of the DL-based methods with conventional features such as the molecular fingerprints (as employed in Ramsundar *et al.* [121]) indicates the improvement yielded by novel featurization approaches. It is also important to note that the performance results given for LR and RF on the MUV results section of Table 4.3 were calculated by Ramsundar *et al.*; however, LR and RF MUV benchmark results we provided in Figure 4.5.b were calculated by us.

We also tested DEEPScreen on the DUD-E dataset and obtained a mean performance of 0.85 area under receiver operating characteristic curve (AUROC). DTI prediction methods utilizing 3-D structural information such as AtomNet [123], Gonczarek *et al.* [232], and Ragoza *et al.* [241] also employed this dataset and reached similar predictive performances. However, their results are not directly comparable with DEEPScreen since these methods utilize both target and ligand information at the input level and reserved some of the targets (along with their ligand information) for the test split during the performance analysis. Also, structure-based methods are usually benchmarked by their success in ranking several docking poses and/or success in minimizing the atomic distances from native binding poses, instead of providing binary predictions as active/inactive. It is important to note that the methods employing 3-D structural features of the target proteins may provide better representations to model DTIs at the molecular level; however, they are highly computationally intensive. Also, 3-D structural information (especially the target-ligand complexes) is only available for a small portion of the DTI space, as a result, their coverage is comparably low, and they generally are not suitable for large-scale

DTI prediction. It is also important to note that DUD-E benchmark dataset is reported to suffer from negative selection bias problem [269], and thus, the results based on this dataset may not be conclusive.

4.4.6 Large-Scale Production of the Novel DTI Predictions with DEEPScreen

DEEPScreen system was applied on more than a million small molecule compound records in the ChEMBL database (v24) for the large-scale production of novel DTI predictions. As a result of this run, a total of 21,481,909 DTIs were produced (i.e., active bio-interaction predictions) between 1,339,697 compounds and 532 targets. Out of these, 21,151,185 DTIs between 1,308,543 compounds and 532 targets were completely new data points, meaning they are not recorded in ChEMBL v24 (the prediction results are available in the repository of DEEPScreen). Apart from this, newly designed compounds that are yet to be recorded in ChEMBL database can also be queried against the modelled targets using the stand alone DEEPScreen models available in the same repository.

We carried out a statistical analysis in order to have an insight about the properties of the compounds predicted for the members of the high level protein families in the large-scale DTI prediction set. For this, an ontology based enrichment test was conducted (i.e., drug/compound set enrichment) to observe the common properties of the predicted compounds. In enrichment analysis, over-represented annotations (in terms of ontology terms) are identified for a query set and ranked in terms of statistical significance[271]. The enrichment tests was done for ChEBI structure and role definitions[272], chemical structure classifications and ATC (Anatomical Therapeutic Chemical Classification System) codes [273], together with experimentally known target protein and protein family information of the predicted compounds (source: ChEMBL, PubChem and DrugBank), functions of these experimentally known target protein and families (Gene Ontology[274]), disease indications of these experimentally known target protein and families (MESH

terms[275] and Disease Ontology[276]). Multiple online tools have been used for this analysis: CSgator [271], BiNChE [277] and DrugPattern[278] .

Since the compounds in the query sets have to be annotated with the abovementioned ontology based property defining terms, we were able to conduct this analysis on a subset of the compounds in the DTI prediction set (i.e., nearly 30,000 ChEMBL compounds for the ChEBI ontology and 10,000 small molecule drugs from DrugBank v5.1.1 for the rest of the ontology types, with a significant amount of overlap between these two). The overall prediction set used in the enrichment analysis was composed of 377,250 predictions between these 31,928 annotated compounds and 531 target proteins. It was not possible to carry out an individual enrichment analysis for the predicted ligand set of each target protein due to high number of targets (i.e., 704). Instead, we analyzed the ligand set predicted for each target protein family (i.e., enzymes, GPCRs, nuclear receptors, ion channels and others) together with an individual protein case study considering the renin protein. For each protein family, the most frequently predicted 100 compounds, each of which has been predicted as active for more than 10% of the individual members of the respective target family are selected and given as input to the enrichment analysis (i.e., a compound should be annotated o at least 38 enzymes in order to be included in the enrichment analysis set of the enzymes, since there are 374 enzymes in total). The reason behind not using all predicted compounds was that, there were high number of compounds predicted for only 1 or 2 members of a target family, which add noise to the analysis when included. ChEMBL ids of the compounds predicted for each target family is given in the repository of the study together with their prediction frequencies.

4.5 Discussion and Conclusion

In this study, we proposed DEEPScreen, a novel deep learning based drug/compound-target prediction system. The major contributions of DEEPScreen to the literature can be listed as:

(i) the idea of using compound images for predicting the interactions with target proteins and employing established convolutional neural network architectures that showed high performance in image recognition/analysis tasks;

(ii) constructing (and open access sharing) a reliable experimental DTI dataset to be used as training/test sets, both in this study and in other future studies. The existing reference DTI datasets are usually small-scale, thus, there is a requirement for high quality large-scale datasets especially for deep learning based model training;

(iii) generating highly optimized, high performance predictive models for 704 different target proteins, each of which was independently trained and optimized with rigorous tests. This approach gave way to a significant performance improvement over the state-of-the-art;

(iv) conducting high number of experiments and data analysis processes in terms of benchmarks / performance tests and comparisons with the state of the art to understand the model/system behavior under different conditions.

(v) publishing the method as an open access tool. DEEPScreen is practical to use since it is composed of independent modules (i.e., each target protein model), where only the model of the target of interest should be downloaded and run to produce predictions;

(vi) executing a systematic large-scale DTI prediction run between 704 targets and 1.3 million drug candidate compounds recorded in the ChEMBL database. Selected examples from the novel predictions has been tested and validated by molecular docking analysis and *in vitro* experiments on cancer cells for potential future drug discovery and repurposing applications.

Considering the main reason why DEEPScreen works better compared to the state-of-the-art DTI prediction approach: molecular descriptors such as fingerprints make assumptions regarding what parts in a molecule are important for target binding and generate feature vectors for storing the information of the presence or absence of these groups (i.e., feature engineering), thus, the information that is deemed unimportant for binding is eliminated. As such, the ML predictor is provided only with a limited piece of information to work with. Besides, it is not possible to generalize these assumptions to the whole DTI space, which is indicated by the limited predictive performance obtained with the conventional approach. By employing 2-D structures generated from SMILES, the system does not make any prior assumptions and just provide a vector displaying the entire molecule with a representation similar to its state in the nature, to let the DCNN to identify the parts necessary for the interaction with the corresponding target protein. Provided with a sufficient number and structural variety of active data points, DEEPScreen was able to learn the relevant interactive properties and provided accurate DTI predictions. Based on the performance results obtained in this study, it is possible to state that the performance improvement of DEEPScreen comes from both using image features and a deep learning approach that is suitable to extract information from images. It is possible that adding the 3-D representations of molecules (i.e., conformational information) to the system would provide a more accurate modelling; however, DCNNs that employ 3-D convolutions are computationally highly intensive, which prevents practical applications at large-scale.

In DEEPScreen, we modelled the interactive properties of each target protein independently in a separate DCNN. This allowed the learning of target specific binding properties during the training process (i.e., the optimization of hyper-parameters and the regular model parameters). In most of the ML method development studies, hyper-parameters are arbitrarily pre-selected without further optimization (especially when there are high number of models as in the case of DEEPScreen), due to extremely high computational burden. However, hyper-parameters are an important part of the model architecture and significantly

contribute to the predictive performance. In this study, we evaluated hundreds to thousands of models for each target, resulting in more than 100,000 model training and evaluation jobs in total (considering the hyper-parameter value options in Table S.1 and their combinations with each other). As a result, a strong computing cluster and extensive levels of parallelization were required to practically run the computational jobs. Whereas, the main advantage of this approach is the elevated predictive performance, which was indicated by the results of the performance tests.

An important concern in ML method development is the problem of overfitting. We employed the neuron drop-out technique, a widely accepted approach for DCNN training, in order to prevent this issue. The results of the independent tests and benchmarking experiments confirmed that overfitting was not a problem for DEEPScreen.

One direction on which DEEPScreen can be improved would be the incorporation of target proteins with only a few known small molecule interactions and the ones without any (i.e., target discovery). DEEPScreen only takes the features of compounds at the input level and treats the target proteins as labels, which allowed ligand predictions for only 704 highly-studied proteins (i.e., the major limitation of DEEPScreen). Within a multi-task modelling approach, targets with only a few known interactions can be incorporated together with the well-studies targets. In this scheme, data augmentation techniques can be incorporated such as the generative adversarial networks to balance the training datasets. To be able to provide predictions for proteins without known interactions, target descriptors may be incorporated at the input level along with compound features, within a chemogenomic modelling approach. Image or graph based structural representations of proteins can be used for this purpose.

CHAPTER 5

MDeePred: MULTI-CHANNEL DEEP CHEMOGENOMIC PREDICTION OF BINDING AFFINITY IN DRUG DISCOVERY

5.1 Chapter Overview

[5] Identification of interactions between bioactive small molecules and target proteins is crucial for novel drug discovery, drug repurposing and uncovering off-target effects. Due to the tremendous size of the chemical space, experimental bioactivity screening efforts require the aid of computational approaches. Although deep learning models have been successful in predicting bioactive compounds, effective and comprehensive featurization of proteins, to be given as input to deep neural networks, remains a challenge. Here, we present a novel protein featurization approach to be used in deep learning-based compound-target protein binding affinity prediction. In the proposed method, multiple types of protein features such as sequence, structural, evolutionary and physicochemical properties are incorporated within multi-channel 2-D vectors, which is then fed to state-of-the-art pairwise input hybrid deep neural networks to predict the real-valued compound-target protein interactions. The method adopts the chemogenomic approach, where both the compound and target protein features are employed at the input level to model their interaction. The whole system is called MDeePred and it is a new method that produce compound-target binding affinity predictions with high-performance, for the purposes of computational drug discovery and repositioning. We evaluated MDeePred on well-known benchmark datasets and compared its performance with

[5] The content of this chapter is under review in *Bioinformatics* journal. Please note that only the parts that I worked on were included from our publication.

the state-of-the-art methods. We also performed *in vitro* comparative analysis of MDeePred predictions with selected kinase inhibitors' action on cancer cells. MDeePred is a scalable method that can handle the available big data from the chemical space, with sufficiently high predictive performance. The featurization approach proposed here can also be utilized for other protein-related predictive tasks.

The source codes, datasets, additional information and user instructions of MDeePred are available at https://github.com/cansyl/MDeePred.

This chapter consists of the parts that I mainly worked in MDeePred. The rest of the conducted research and analysis can be reached from our publication. My specific contributions in MDeePred are listed below:

- Development and design of the proposed method;
- Implementation of the overall system;
- Investigation of state-of-the-art methods and benchmarking datasets;
- Training of models for DREAM Kinase-Drug Prediction challenge and running models on target datasets;
- Implementation of scripts for the analysis and discussions.

5.2 Introduction

The identification of new compounds with high binding affinities for intended target proteins is an important step in early drug discovery. Traditionally, drug discovery starts by target protein selection, which is followed by high-throughput assays to screen candidate interacting compounds. However, conducting high-throughput screening experiments is not feasible for the massive compound and protein space as it is a time-consuming and expensive process. Therefore, several computational methods have been proposed with the aim of providing accurate binding affinity predictions. The predicted bioactive small molecules (i.e., ligands) with desired binding affinities (i.e. strong binders) can then be used as drug candidates for further experimental validation against their target proteins (i.e., receptors).

Machine learning approaches for drug-target binding affinity prediction can be categorized into two groups in terms of the input representation type: feature-based and similarity-based. In the feature-based approach, features of each compound or target protein are individually extracted and the extracted features are represented as a vector [13], [14], [56], [81], [85], [228]. In the similarity-based approach, the similarity for a pair of compounds or for a pair of target proteins is calculated and often utilized within a network based link prediction approach [73], [76], [79], [279]–[281]. Drug-target binding affinity prediction methods can also be categorized based on the type of the prediction output as either qualitative values, which can be independent classes (i.e., classification methods) or quantitative values, which are the actual binding affinity values (i.e., regression methods). A classification method may provide binary output for an input sample as active/interacting or inactive/non-interacting. One important drawback of the classification approach is that the binding affinity values are not estimated, and the knowledge of whether a compound-target pair is active or inactive alone is not always informative. On the other hand, predicting actual binding affinity values between compounds and target proteins is more informative in terms of interpretability of results, albeit being a more challenging problem. Computational drug discovery methods can also be classified in terms of the employed featurization approach, where the conventional methods only use the ligand features for the modelling [7]. A rather new paradigm introduced the chemogenomic approach, where both compounds and target proteins are featurized and fed to the prediction model at the input level [282]–[284]. Recently, a few studies aimed to aid the drug discovery research by providing binding affinity value predictions [284]–[287]. The results reported in these studies indicated that there is a requirement for new approaches with improved performance before these methods can be integrated to actual drug discovery pipelines.

Thanks to the advances in computing and the exponential growth of available data, deep learning algorithms, have been successfully applied in several fields such as speech recognition, image analysis, bioinformatics, cheminformatics and computational chemistry [7], [114], [115], [177], [288]–[290]. Among the deep

learning approaches for drug-target binding affinity prediction, DeepDTA is a non-structure based binding affinity value prediction method that employs the chemogenomic approach, where both drug and target protein features are used at the input level, together with deep neural networks [284]. Encodings for amino acids in protein sequences and characters in SMILES strings are generated by assigning unique integer values to each of them. Drugs and proteins are represented by 1-D numerical feature vectors based on these encodings. Then, the feature vectors are fed to convolutional neural netwoeks (CNN), with 1-D convolutional and feed-forward layers. The output of DeepDTA is the predicted binding affinity value of the input drug-target protein pair. DeepDTA is compared with state-of-the-art methods. Among the methods that use a machine learning method and also the similarity-based approach, SimBoost has been developed to predict binding affinity values [291]. In SimBoost, three types of features are calculated: individual features for each drug and target protein and two types of network-based features. The extracted features are then fed to the gradient boosting machine method to predict binding affinities of drug-target protein pairs.

Performance evaluation is an important issue for the prediction of binding affinity in drug discovery. MoleculeNet is a benchmarking platform designed for evaluating and testing computational methods for molecular property predictions [234]. It includes multiple benchmarking datasets and the evaluation results for several machine learning algorithms (e.g. deep neural networks, random forest) and feature types (e.g. extended connectivity fingerprints-ECFP, graph convolutional features, grid featurizer, and etc.). The benchmarking datasets, along with the training/validation/test splits, are also provided.

So far in the literature, chemogenomics based binding affinity prediction systems incorporated the target proteins in their system only using the information provided by the fact that a protein sequence is composed of a special order of 20 different characters. However, there is a wealth of accumulated knowledge regarding the structural, evolutionary and physico-chemical properties of these proteins, which can be incorporated in a binding affinity prediction model. The hypothesis in this study

is that, drug-target protein interactions can be modelled in a more successful manner with the inclusion of various attributes of proteins, and this would be reflected as a prediction performance increase in this field. Convolutional neural networks (CNN) have outperformed state-of-the-art methods, especially in the field of image analysis, where inputs are generally the 2-D images to be classified. In a 2-D color image, there are 3 input color channels (RGB channels) and therefore, 3 matrices containing intensity values for each pixel. Each input channel can be considered as a feature of the corresponding image. In fact, one can generate other input channels, possibly more than three, where each input channel represents a different type of feature belong to the input sample.

In this study, we propose Multi-channel Deep Chemogenomic Predictor for Binding Affinity (MDeePred), a novel compound-target protein interaction prediction system, by pursuing the idea of constructing multiple channels that represent the input proteins from different aspects. MDeePred follows the chemogenomic modeling approach, where both protein features and compound features are used as input to the system. For this, we defined multiple feature matrices (as our input channels) to represent protein sequences. For each feature matrix, we calculated a specific property of each amino acid pair in that protein sequence, such as mutational probabilities, probabilistic residue-residue contact distances, physicochemical properties and a simple encoding by enumerating each amino acid pair. We also represented compounds using ECFP4 circular fingerprints. We first fed the multi-channel feature matrices to a convolutional neural network, which constitutes the target protein side of the model and the compound feature vectors to a separate feed-forward deep neural network, which constitutes the compound side of the model. At a second stage of MDeePred, the output of the target protein side is concatenated with the last layer of compound side. The final output of MDeePred is a single neuron that produces a prediction for binding affinity value of the input drug-target protein pair, via a regressor. MDeePred was implemented with PyTorch framework [292]. The source code, datasets, experimental settings and instructions about usage of the system are available at the GitHub Repository https://github.com/cansyl/MDeePred.

5.3 Materials and Methods

5.3.1 Data

We used three popular benchmarking datasets to assess the performance of MDeePred and to compare its results with those of the state-of-the-art methods. Davis et al. performed screening experiments to determine the binding affinity values of 68 kinase inhibitors against 442 protein kinases in human catalytic protein kinome [139]. This dataset is called Davis Dataset and there are 30,056 bioactivity values (Kd values) which span all compound-target protein pairs. In this dataset, about 70% of the bioactivity values (20,931 out of 30,056) are recorded as 10 μM. These binding affinity values indicate compound-target protein pairs for which no binding were observed in the primary screen, where authors screened ligands against the panel at a single concentration of 10μM [139]. As a result, it is not suitable to use these data points in a regression based prediction system. In addition, incorporating these data points into the training datasets may cause biased predictive models and misleading performance results. For example, a predictive model can achieve a low mean squared error value by predicting all binding affinity values around 10μM. Hence, we filtered out the data points with 10 μM bioactivity values and the new dataset, containing 9,125 binding affinity values, is named the Filtered Davis Dataset. PDBBind is a comprehensive resource of experimentally measured binding affinity data for protein-ligand complexes, which are derived from the Protein Data Bank [293]. It contains binding affinity values and 3-D structures of protein-ligand complexes. PDBBind Refined Dataset is a subset of PDBBind and a widely-used benchmark set [294]. There are 3,706 receptor-ligand complexes and their binding affinity values in PDBBind Refined Dataset.

5.3.2 Protein Feature Matrices

The aim here is to create numerical representations of protein sequences, to be used as the input to the predictor model. As the base input channel of the convolutional neural network, protein sequences are represented by encoding matrices. Let A represents the set of 20 standard amino acids. We can then define the cartesian product of A by itself:

$$A \times A = \{\langle a, b\rangle;\ a, b \in A\}$$

Let M be set of integers which is defined as

$$M = \{k \in Z^+ \mid k \leq \left(\frac{|A|^2 - |A|}{2} + |A|\right)\}$$

We can define a surjective mapping function $f : A \times A \rightarrow M$ such that pairs $\langle a, b\rangle$ and $\langle b, a\rangle$ are mapped to a unique integer k in set M. For example, $\langle A, A\rangle$ is mapped to 1, ordered amino acid pairs $\langle A, R\rangle$ and $\langle R, A\rangle$ are both mapped to 10, and so on. If a protein sequence S is represented as $S = \langle s_1, s_2, \ldots, s_n\rangle$ where s_i represents i^{th} amino acid in the sequence, we can then construct an encoding matrix N whose size $n \times n$, where rows and columns are amino acids of S from the N terminus to the C terminus. We can use the mapping function f to determine the value of the element N_{ij} $1 \leq i, j \leq n$ as follows:

$$N_{ij} = f\big(\langle s_i, s_j\rangle\big)$$

Following this methodology, we constructed an encoding matrix for each protein sequence. The diagonal elements of the encoding matrix represent the sequence itself. The remaining elements represent the amino acid matches in different positions of the corresponding protein sequence. Since protein sequences have varying lengths, we selected a maximum allowed length value and truncated the rest of the proteins, in order to have fixed-size matrices. For the maximum length, we tested 500 and 1000 amino acids. For protein sequences shorter than the maximum length, the matrices were zero padded. The idea behind the protein encoding matrix

proposed here is similar to the 1-D protein representation idea described in previous methods such as DeepDTA.

AAindex is a database of indices and matrices that represent physicochemical and biochemical properties of amino acids and pairs of amino acids (Kawashima et al., 2008). In this study, we generated additional input channels using numerical 20x20 amino acid matrices from the AAindex database, where each channel represents a different property of the protein sequence. These feature matrices were generated pursuing the same methodology as described above, with the difference that we used the values from the corresponding AAindex amino acid matrices. This way, we were able to represent protein sequences as multi-channel input features. We used ZHAC000103 (the first protein structure representing feature), BLOSUM62 (representing the evolutionary information), GRAR740104 (physico-chemical part) and SIMK990101 (the second protein structure representing feature) matrices from the AA index database. The information about these matrices are given in Table 5.1. Inclusion of these four matrices make MDeePred stand out from the previous studies in this field. Construction of feature matrices for sample partial protein sequences are demonstrated in Figure 5.1.

Table 5.1 Amino acid matrices selected from the AAindex database to generate the input channels.

AAindex DB Identifier	*Feature Type*	*Name of the Matrix*	*Reference*
ZHAC000103	Structure-Based	Environment-dependent residue contact energies	[295]
BLOSUM62	Evolutionary	Amino acid substitution	[296]
GRAR740104	Physicochemical	Chemical distance	[297]
SIMK990101	Structure-Based	Distance-dependent statistical potential (contacts within 0-5 Angstroms)	[298]

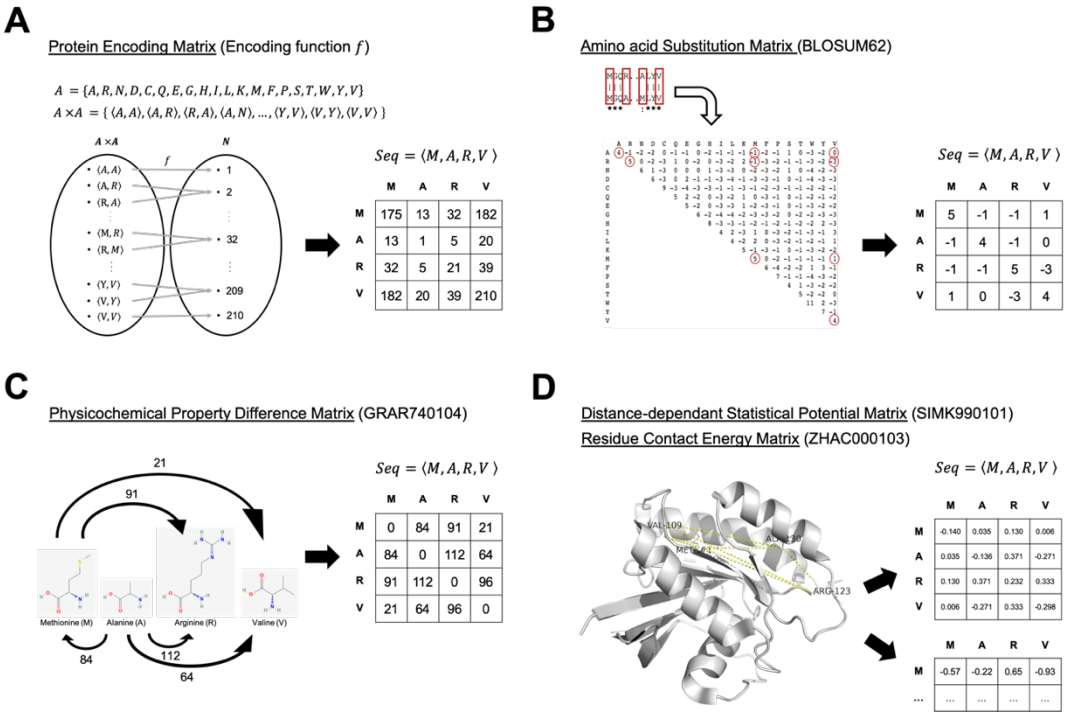


Figure 5.1. Construction of multi-channel input protein feature vectors for an example of 4 amino acid peptide sequence: "M A R V", using the protein encoding matrix (a), substitution scoring matrix (b), physicochemical property difference matrix (c), statistical potential and residue contact energy matrices (d). For the channels except the protein encoding, corresponding 20-by-20 AAindex matrices are used to construct 500-by-500 or 1000-by-1000 2-D feature matrix of the input proteins.

5.3.3 Pairwise-input Hybrid Neural Network

We developed a pairwise-input hybrid neural network for chemogenomic modeling. On the target protein side, a CNN is used and protein feature matrices are fed as input channels to CNN. CNN consisted of two convolutional + pooling layers, which were followed by an inception module. On the compound side, we generated ECFP4 fingerprints using the SMILES strings of compounds, which were fed to a feed-forward neural network. Protein input CNN and compound input feed-forward neural network constitute the first stage of the system. The output of inception

module (of the protein side) was flattened and it is concatenated with the last layer of compound-side and the concatenation layer was followed by two fully connected layers, constituting the second stage of the system. The output is a single neuron that gives a prediction for the binding affinity value of the input drug-target protein pair using a regressor (Figure 5.2). Rectified Linear Unit (ReLU) was used as the activation function in both convolutional and feed forward layers [299]. After each layer, batch normalization was applied preceding ReLU activation function. The objective of the model is to minimize the mean squared error (MSE) during training which measures how much the prediction values differ from the real binding affinity values.

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

where Y_i stands for the real binding affinity values for the i^{th} input pair and P_i represents the prediction value for the same input pair.

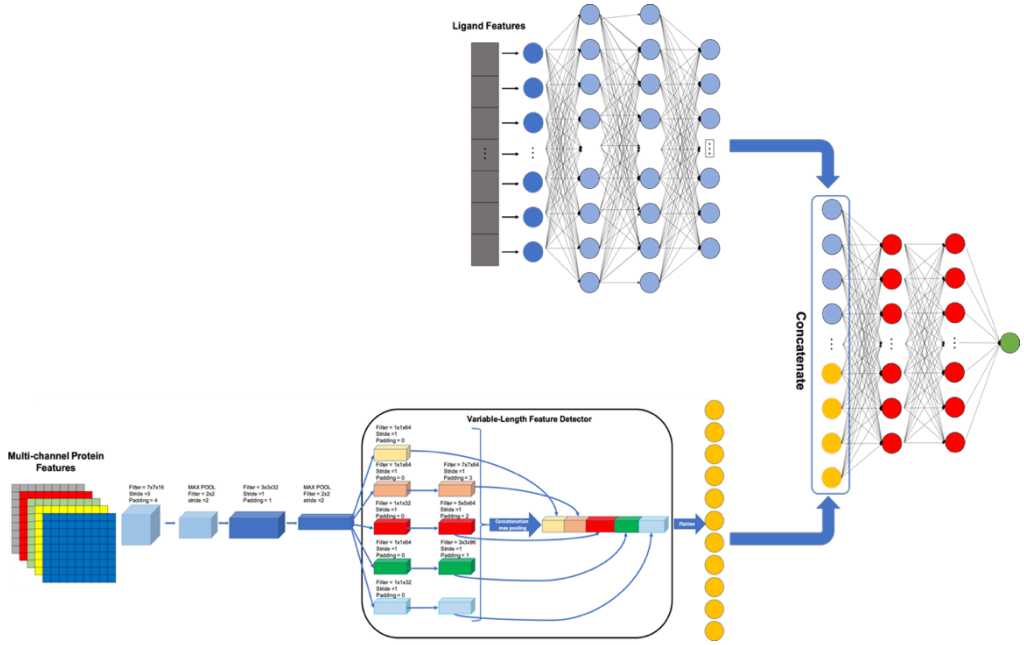


Figure 5.2. Construction of multi-channel input protein feature vectors for an example amino acid sequence. For channels 3, 4 and 5, the corresponding 20-by-20 AAindex matrix is used to construct the 2-D feature, similar to the substitution (BLOSUM62) channel construction.

5.4 Results

We evaluated MDeePred under different settings, in multiple tests. First of all, we measured the performance of MDeePred using different channel combinations to evaluate the effectiveness of the multi-channel protein representation approach. Second, we compared the finalized version of MDeePred with the state-of-the-art methods in the field, which were CGKronRLS, DeepDTA, SimBoost and four other methods in the MoleculeNet Benchmarking platform [234], [284], [291], [300], using 3 different benchmark datasets. These methods, evaluation metrics and the experimental settings are explained in the following subsections.

5.4.1 Performance Evaluation Metrics

To evaluate the performance in binding affinity prediction as a continuous value, we used concordance index (CI) measure [301] and the Spearman rank correlation. CI (over a set of paired data) measures the probability of two randomly selected compound-target protein pairs with different binding affinity values to be in the correct order.

$$CI = \frac{1}{Z} \sum_{\gamma_i > \gamma_j} h(f_i - f_j)$$

where f_i and γ_i represents the predicted and real binding affinity values for the i^{th} pair, respectively. Z is a normalization constant which is equal to the number of drug-target protein pairs and $h(x)$ is a step function defined below:

$$h(x) = \begin{cases} 1.0 \; if \; x > 0 \\ 0.5 \; if \; x = 0 \\ 0.0 \; if \; x < 0 \end{cases}$$

We also considered the problem of binding affinity prediction as a binary classification problem (i.e., active/binder vs. inactive/non-binder) based on four different bioactivity cut-off values (10 μM, 1 μM, 100 nM and 30 nM) and calculated two other metrics to assess the performances of the systems under these cut-off

values. These evaluation metrics were Matthews Correlation Coefficient (MCC) and Area Under Precision-Recall Curve (AUPRC). The real values bioactivity predictions below the selected cut-off values were considered to be active/binder predictions and the ones above the cut-off values were considered to be inactive/non-binder predictions. Precision, recall and MCC metrics are defined below:

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

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

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

5.4.2 Training, Validation and Test Settings

We employed two settings (i.e., Setting-1 and Setting-2) for training, validation and test of the methods, based on the benchmarking datasets. Setting-1 and Setting-2 were derived from DeepDTA and MoleculeNet studies, respectively, and these settings were applied exactly as described in the corresponding studies for a fair comparison. In Setting-1, a nested cross-validation was employed, where the whole dataset was randomly divided into six parts; five of them were used for 5-fold cross validation and the remaining part was used as the independent test dataset. The hyper-parameter values of the models were estimated using 5-fold cross validation and the final performance of the system was evaluated using the independent test set. Evaluation of the methods was carried out with Setting-1 on Davis Dataset and Filtered Davis Dataset. In Setting-2, we employed time-splitting and divided the dataset into three subsets as training, validation and test datasets. Data points coming from earlier publications (up to year 2011) were used as the training dataset, while newer data points (from the year 2012) made up the validation dataset. Finally, the latest data points (year 2013 and 2014) were chosen as the test dataset. The details

are explained in the MoleculeNet article [234]. The evaluation of the methods on the PDBBind Refined Dataset was carried out under the Setting-2.

5.4.3 Deep Neural Network Architectures and Their Hyper-Parameters

In the compound side (i.e., the feed-forward network) of the MDeePred system, 2 and 3 hidden layers were tested with varying number of neurons on each layer. The information about the network architecture of the protein side (i.e., the convolutional network) of MDeePred is given in Table 5.2. Several different values (including their combinations) were tested for the hyperparameters such as the learning rate, number of neurons in each layer, the drop-out rate and etc., to obtain optimal predictive models. Also, the information on the exact network architectures and hyper-parameters are provided in our GitHub repository at https://github.com/cansyl/MDeePred.

Table 5.2 Layers and parameters of the CNN architecture of the protein side. The output of each operation is the input of the following layer.

Operation	Filter size / Stride / Padding	Output Channel	Input
Convolutional	7×7 / 3 / 4	16	5× 500× 500
Pooling	2×2 / 2 / 0	NA	16×168× 168
Convolutional	3×3 / 1 / 1	32	16×84× 84
Pooling	2×2 / 2 / 0	NA	32×84× 84
Inception Module	NA	320	32×42× 42
Pooling	2×2 / 2 / 0	NA	320×42× 42

5.4.4 Evaluation of Protein Feature Matrices

To select the feature matrices for the input and to analyze their effects on the performance, we trained MDeePred with different combinations of input channels over the same set of hyperparameters on PDBBind Refined Dataset. For this, we first trained MDeePred using only the base encoding matrix. Subsequently, new models were trained by adding different input channels and the performances of these models were calculated. Same sets of hyperparameters were used for a fair comparison and the obtained lowest mean squared error values were considered. The results (given in the GitHub Repository) showed that adding input channels improved the performance, compared to only using the proposed encoding matrix. The best performance is obtained with five-input-channeled network where the first channel was the proposed encoding matrix (please refer to Method section) and the remaining four channels were generated based on the amino acid matrices derived from AAindex database (Table 5.1). With the observation of these results, we decided to employ the full five-channel protein representation to construct MDeePred. The whole set of amino acid matrices that were used to construct the input channels, the test performance results for different channel combination (based on different hyperparameters) are given in the GitHub Repository.

5.4.5 Performance Comparison with State-of-the-art Methods

Performance evaluation is an important topic for the problem of binding affinity prediction in the field of computational drug discovery. With aim of making a comprehensive evaluation, the performance of MDeePred was compared with CGKronRLS, DeepDTA, SimBoost, Grid-RF (MoleculeNet), Grid-DNN (MoleculeNet), ECFP4-RF (MoleculeNet) and ECFP-RF (MoleculeNet) methods, which can be considered as the state-of-the-art. The datasets used for the performance comparison were the Davis dataset, Filtered Davis dataset and the PDBBind Refined dataset. MDeePred and state-of-the-art methods were trained

using the same experimental settings for a fair comparison, which means that the data points in training, validation and test datasets were exactly the same. The real valued binding affinity value predictions obtained from different methods were evaluated using the concordance index (CI), mean squared error (MSE) and Spearman rank correlation. We also calculated classification performances (i.e., MCC and AUPRC) of these methods based on different binarization thresholds, all of which were previously defined as important activity thresholds for different target protein families [302]. For PDBBind Refined dataset analysis, we directly used the predictions for train/validation/test splits, which were provided by the authors of the MoleculeNet study.

For the performance assessment, we first used Davis dataset and the filtered Davis dataset under the setting-1 (please refer to Train-validation-test Settings sub-section under the Methods section). The performance calculation procedure was applied for MDeePred, CGKronRLS, DeepDTA and SimBoost methods on the Davis dataset. The results are available in Table 5.3, where higher values indicate better performance except for the MSE. Here, MDeePred achieved the best performance in terms of all metrics on the Davis dataset.

We also trained MDeePred, CGKronRLS and DeepDTA methods using the Filtered Davis dataset and the results are given in the Table 5.4. In this analysis, we could not train SimBoost, as the implementation were not suitable to train the method on different datasets and it also relies on pre-calculated inputs that were not available for the Filtered Davis dataset. As it can be observed from the performance results, MDeePred and CGKronRLS outperformed DeepDTA in all of the evaluation metrics. The performance difference becomes more pronounced in the lower bioactivity thresholds as DeepDTA's performance drops significantly as the bioactivity threshold is decreased. When MDeePred's performance is compared with that of CGKronRLS, it is observed MDeePred achieves the lowest MSE and it also gets better results in the majority of the cases.

We used setting-2 for training the methods on the PDBBind refined dataset and the performance results on the independent test split are given in Table 5.5. When we consider the overall performance results, MDeePred was the best performing method. The models that were trained with the grid features (Grid-RF, Grid-DNN) had better performance compared to the models trained with the molecular fingerprint-based features. Considering the classification performances based on different bioactivity thresholds, it was observed that the performances of Grid-RF and Grid-DNN methods significantly decreased as the bioactivity threshold is decreased. For example, the MCC score of Grid-RF was almost random (0.091) at 30 nM threshold, whereas MDeePred's MCC score at the same threshold was 0.453.

Table 5.3 Davis dataset average performance test results on the independent test fold for MDeePred, CGKronRLS, DeepDTA and SimBoost methods (under setting-1). Standard deviations are given in parenthesis and the best results are highlight with bold font.

Method	CI	MSE	Spearman	Average AUPRC	MCC (30nM)
MDeePred	**0.886 (0.001)**	**0.254 (0.002)**	**0.69 (0.003)**	**0.744 (0.003)**	**0.626 (0.012)**
CGKronRLS	0.873 (0.001)	0.284 (0.003)	0.671 (0.002)	0.724 (0.004)	0.604 (0.011)
DeepDTA	0.867 (0.006)	0.310 (0.016)	0.665 (0.007)	0.679 (0.016)	0.576 (0.030)
SimBoost	0.876 (0.002)	0.284 (0.004)	0.677 (0.003)	0.705 (0.004)	0.557 (0.020)

Table 5.4 Filtered Davis dataset average performance test results on the independent test fold for MDeePred, CGKronRLS and DeepDTA methods (under setting-1). Standard deviations are given in parenthesis and the best results are highlight with bold font.

Method	CI	MSE	Spearman	Average AUPRC	MCC (30nM)
MDeePred	0.733 (0.004)	**0.545 (0.007)**	0.618 (0.009)	**0.803 (0.006)**	0.585 (0.010)
CGKronRLS	**0.740 (0.003)**	0.591 (0.015)	**0.643 (0.008)**	0.773 (0.010)	**0.617 (0.029)**
DeepDTA	0.653 (0.005)	0.866 (0.028)	0.430 (0.013)	0.529 (0.018)	0.208 (0.035)

Table 5.5 PDBBind refined dataset performance results for MDeePred and the MoleculeNet benchmarking methods (under setting-2).

Method	CI	MSE	Spearman	Average AUPRC	MCC (30nM)
MDeePred	**0.754**	**2.494**	**0.681**	**0.768**	**0.453**
Grid RF	0.729	3.4	0.634	0.723	0.091
Grid DNN	0.67	3.616	0.505	0.643	0.314
ECFP4 RF	0.657	3.207	0.483	0.638	0.186
ECFP RF	0.608	5.255	0.344	0.545	0.138

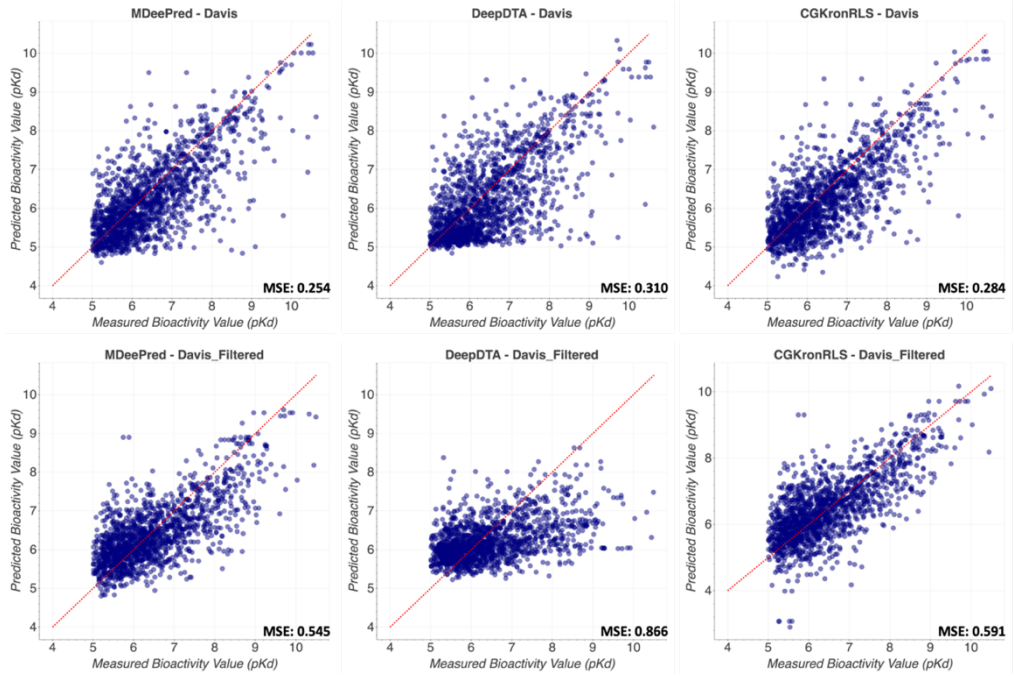


Figure 5.3. Measured (true) vs. predicted binding affinity plots for MDeePred, DeepDTA and CGKronRLS, on Davis and the filtered Davis datasets

Figure 5.3 presents the predicted bioactivity values against the measured bioactivity values of MDeePred and DeepDTA for Filtered Davis Dataset. Similar plots of MDeePred, DeepDTA, SimBoost and four methods from MoleculeNet framework for Davis Dataset, Filtered Davis Dataset and PDBBind Refined Dataset are available in the GitHub Repository. A perfect predictive model is expected to provide predictions over the $y=x$ line (red line) in these plots. As it can be observed in Figure 5.3, the points in MDeePred's plot is much more aligned with the red line than those in DeepDTA's plot. On the Davis Dataset, DeepDTA's predictions is highly biased towards pKd value of 5.0 whereas MDeePred and SimBoost is more robust (shown in the GitHub Repository). For PDBBind Refined dataset, MDeePred's prediction performance is better than the four methods of MoleuculetNet in general, and it can be seen that the prediction accuracy of the other methods is getting worse as the actual bioactivity value decreases (GitHub Repository).

5.4.6 *in vitro* Comparative Analysis of MDeePred Pre-dictions with Selected Kinase Inhibitors' Action on Cancer Cells

We have selected small molecule inhibitors (Staurosporine, Rapamycine, NVP-BEZ235 and Alsterpaullone) to perform a comparative analysis between MDeePred bioactivity predictions and the effect of these compounds on two human cancer cell lines Huh7 and Mahlavu. Initially, we identified IC_{50} values of the selected compounds on Huh7 and Mahlavu cell lines. We then performed RNAseq whole-transcriptome profiling which were used to globally identify the genes that are differentially regulated in the presence of selected small molecule inhibitors (DEG data is given in the GitHub repository). In parallel we trained an MDeePred kinase model using an in-house kinase dataset derived from ChEMBL database. An initial training dataset was created by filtering and preprocessing bioactivities from ChEMBL database which resulted in 139,826 bioactivities over 346 protein kinases and 90,518 ligands. ECFP4 fingerprints of ligands were then generated and clustered based on the Tanimoto coefficient value of 0.7 in order to avoid chemical series bias during training and evaluation of the model. The final dataset consisted of 49,277 bioactivities which spanned 346 protein kinases and 29,214 ligands. Subsequently, the final dataset was randomly divided into training (64%), validation (16%) and independent test (20%) and the performance was calculated on the independent test dataset which is given in Table 5.6.

Table 5.6 The performance results of the MDeePred kinase model.

Model	CI	MSE	Spearman	Average AUPRC	MCC (30nM)
MDeePred-kinase	0.779	0.693	0.747	0.815	0.567

Using this newly trained MDeePred kinase model the binding affinities for Staurosporine, Rapamycine, NVP-BEZ235 and Alsterpaullone toward 346 protein

kinases were predicted (given in the Github repository). The experimental results and related discussion are available in MDeePred manuscript.

5.5 Discussion and Conclusion

In this study, we presented a novel featurization approach to represent protein sequences as numerical matrices, based on their physical, chemical and biological properties. By using the proposed featurization approach, we tackled the problem of predicting bio-interactions at large scale in the framework of computational drug discovery. More precisely, we developed a chemogenomics-based method, MDeePred, to predict the binding affinities between drugs/drug like small molecule compounds and target proteins based only on their most common representations; amino acid sequences for proteins and SMILES strings for compounds. In order to achieve better generalizations, we preferred to present the target protein and the compound as a pair, at the input level of MDeePred. The output of MDeePred is a quantitative value representing the binding affinity between the input compound and the target protein (equivalent to IC50). MDeePred contains a hybrid deep neural network structure with a regressor attached to the output layer, to process the input feature matrices and to provide the real-valued prediction result.

Our featurization approach is expressed as multiple channels that represent the input proteins from different aspects and these channels constitute the protein input of MDeePred. Performance tests on these channels indicated that the model with input channels of amino acid substitution scores, physicochemical property comparisons and 3-D structural features together with the baseline encoding scheme performed the best, which supported our initial claim of representing a protein sequence from multiple aspects would lead to better modeling. It is also important to note that, the architecture of MDeePred is suitable to take in additional channels, that is input protein feature matrices, as desired. Therefore, MDeePred is able to draw out complex features from multiple and diverse representations of proteins. We believe that the described feature matrix construction approach is a significant improvement

for protein sequence representation in the field of chemogenomics-based drug/compound-target protein interaction prediction.

We evaluated MDeePred with comparisons against other binding affinity prediction methods. For this, MDeePred and 7 state-of-the-art methods were rigorously assessed on the Davis dataset, filtered Davis dataset and the PDBBind refined dataset. The results of MDeePred were in general more consistent and better than those of other 7 state-of-the-art methods. Furthermore, the performance of MDeePred was stable for different bioactivity thresholds. Improvements over the existing literature, brought by this study can be listed as follows:

- Featurization: The novel protein featurization approach proposed in this study, together with the proposed predictive modeling approach, can be used for other protein related automated annotation tasks in bioinformatics, such as the prediction of protein-protein interactions or the prediction of protein functions.
- Deep learning architecture: MDeePred extracts complex representations separately from target proteins and compounds, and then merges these complex representations to infer the binding properties of these pairs. The hybrid deep neural network model is carefully designed considering its architecture and fine-tuned in terms of its parameters, so that it usually achieves a better prediction performance compared to the state-of-the-art methods in recent literature.
- Open access repository: The source code, datasets, results and user instructions of MDeePred are available at: https://github.com/cansyl/MDeePred. Furthermore, trained prediction models are accessible, especially to predict binding affinity values between various compounds and protein kinases. Also, it is easy to train other target family-based models using the provided code.

We believe that there is still room for improvement for the prediction of drug/compound-target protein binding affinities at large scale, especially considering

novel ways of representing input samples and the cutting-edge machine learning algorithms. For further analysis, inference and prediction; the output of the drug/compound-target protein interaction prediction methods can provide essential information to represent complex relations between drugs/compounds, genes/proteins, pathways and diseases for systems such as biological knowledge graphs [303], and to find interactions between the characteristics of drugs and patient specific disease cell lines [304] for novel precision medicine applications.

CHAPTER 6

iBioProVis: DRUG-TARGET INTERACTION PREDICTION WITH CONVOLUTIONAL NEURAL NETWORKS USING 2-D STRUCTURAL COMPOUND REPRESENTATIONS

6.1 Chapter Overview

iBioProVis is an interactive platform for visual analysis of the compound bioactivity space in the context of target proteins, drugs, and drug candidate compounds. iBioProVis framework takes target protein identifiers and, optionally, compound SMILES as input, and uses the state-of-the-art non-linear dimensionality reduction method t-Distributed Stochastic Neighbor Embedding (t-SNE) to plot the distribution of compounds embedded in a 2-D map, based on the similarity of structural properties of compounds and in the context of compounds' cognate targets. Similar compounds, which are embedded to proximate points on the 2-D map, may bind the same or similar target proteins. Thus, iBioProVis can be used to easily observe the structural distribution of one or two target proteins' known ligands on the 2-D compound space, and to infer new binders to the same protein, or to infer new potential target(s) for a compound of interest, based on this distribution. iBioProVis also provides detailed information about drugs and drug candidate compounds through cross-references to widely used and well-known databases, in the form of linked table views

This chapter consists of the parts that I mainly worked in iBioProVis. The rest of the conducted research and analysis can be reached from our publication. My specific contributions in iBioProVis are listed below:

- Investigation of embedding algorithms such as t-SNE, UMAP and PCA;

- Application and analysis of the embedding results for t-SNE, UMAP and PCA based on ECFP4 fingerprints;
- Investigation of visualization frameworks such as Plotly, Bokeh, Seaborn and deciding the framework to be used based on the analysis;
- Investigation of other databases such as Clinical Trials, IntAct, PubChem, DrugBank and getting cross-references from them;
- Implementation of backend development of iBioProVis based on t-SNE and UMAP algorithms;
- Implementation of first version of iBioProVis using Bokeh framework and making improvements on the tool.

6.2 Introduction

The ChEMBL database (version 25) has 1,879,206 distinct compounds with 12,482 target proteins and 15,506,670 reported bioactivities [225]. Even if only the data in ChEMBL are considered, there are more than 11 billion possible compound-target protein pairs to be tested *in vitro* experimentally. Unfortunately, public databases or datasets have limited coverage as only partial information is available regarding the compound-target interaction space, mainly due to high costs and labor requirements associated with large-scale screening experiments. Therefore, prior knowledge about the eventual target proteins or cellular signaling events in which a small molecule is involved in becomes crucial for novel drug-target discovery [305]. Furthermore, the representation of drugs and their targets in databases lack the comparative holistic view of the molecular action on multiple targets and structural similarity of the compounds.

A few number of studies have recently become available to visualize the chemical space and the compound bioactivity space [305]–[308]. Karlov et al. [306] and Drug Discovery Maps [307] made visualization tools available, only for pre-computed datasets. In webDrugCS [308], visualization is performed by PCA which is a linear and global method and PCA is known to miss non-linear and local relations among

the input drug molecules. Another study, Gaspar et al. only present their results and no tool is made available [309]. There are no target proteins in CheS-Mapper [305], however, the user can apply clustering on the compounds to observe their groupings.

We describe a framework called iBioProVis, which uses a map-based method to embed active compounds, in the context of their cognate target proteins, as points onto a real coordinate-based 2-D space, based on the structural descriptors of the active compounds. iBioProVis allows the interactive visualization of the embeddings. The input to iBioProVis is a set of ChEMBL identifiers or UniProt identifiers for target proteins and the output is the 2-D embedding of the active compounds of these proteins. By looking at the distribution of active compounds as points in this embedding, the user can infer that the compounds that are close to each other may possess similar protein target characteristics. We use the extended connectivity fingerprint [247] with bond diameter four (ECFP4) as the compound descriptor and t-Stochastic Neighbor Embedding (t-SNE) to generate the 2-D embeddings. We also provide a reliable compound-target bioactivity measurement dataset, which is a carefully processed and filtered subset of ChEMBL (v25), to be used with iBioProVis.

6.2.1 Material and Methods

iBioProVis has its own in-house dataset processed and filtered from ChEMBL (v25), which originally contains a total of 15,506,670 data points (i.e., bioactivity measurements) [225]. After the application of several filtering and pre-processing steps to generate the iBioProVis compound-target protein dataset, the number of bioactivity measurements was reduced to 890,886 which contains 3,803 unique target proteins and 581,442 unique compounds. iBioProVis embedding operations are applied on this filtered dataset. Upon a user submission of target protein identifier(s), iBioProVis first extracts ECFP4 for the compounds of the given target protein(s), to be used as compound feature vectors. The tool then generates a distance

matrix for the given compounds, based on the Tanimoto coefficient whose equation is given below:

$$Tanimoto\ Coefficient = \frac{c}{a + b + c}$$

In the above equation; given two compounds, compound A and compound B, and their fingerprints D_A and D_B: a is the number of dimensions set to 1 in D_A but not in D_B. b is a is the number of dimensions set to 1 in D_B but not in D_A. c is the number of dimensions set to 1 in both D_A and D_B. The distance matrix becomes the input to the t-SNE algorithm which produces the 2-D embeddings of the compound feature vectors [310]. Finally, these 2-D embeddings are plotted as a scatter plot and the point that corresponds to each compound is color-labelled based on the target protein that the compound is reported to bind to. It is also possible to give the representations of drugs or compounds of interest in SMILES notations during the input phase, to obtain their 2-D embeddings along with the binders of the given target proteins. Once the embedding process is completed and displayed, the user is able to select a set of compounds on the constructed plot and observe their ChEMBL identifiers and the target proteins that they actively bind to. The steps that are followed to generate the embeddings of compounds in iBioProVis are given in Algortihm 1.

Several cross-references to widely used and well-known biological databases are also provided so that the user can easily relate the entities and navigate to those databases by clickable links. The cross-referenced databases are UniProt, IntAct, PubChem, DrugBank and Clinical Trials. The Bokeh library is employed to generate interactive and user-friendly visualizations [311].

Algorithm 1: iBioProVis algorithm to produce compound embeddings

User Input: A set of target protein ChEMBL identifiers and optionally SMILES strings of a set of user-defined compounds

Output: 2-D embedding plot of color-labeled compounds

1: Get ChEMBL identifiers of active compounds of input target proteins

$C = \{c_1, c_2, \ldots, c_N\}$

2: Get SMILES strings of compound set C $S = \{s_1, s_2, \ldots, s_N\}$

3: Calculate ECFP4 descriptors as feature vectors for set S $F = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N\}$

4: Calculate distance matrix $\boldsymbol{D}$ for all feature vectors in F using Tanimoto coefficient

5: Feed $\boldsymbol{D}$ to t-SNE algorithm as input to get 2-D embeddings of compounds

$E = \{(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)\}$

6: Plot embeddings E and apply color labeling based on the target proteins that compounds actively bind to.

6.2.2 Case Study

A sample embedding is given in Figure 6.1. The active compounds of the target proteins AGPAT2 (CHEMBL4772) and PLK2 (CHEMBL5938) are embedded on the 2-D space, where the active compounds of AGPAT2 are colored in blue and the active compounds of PLK2 are in green. Additional user-input compounds are represented by the red color and drugs (approved and experimental drugs found in the DrugBank database) are represented by diamond shapes. When a user selects a set of compounds, the information about these compounds and their target proteins is shown in two different tables (side table: compounds, bottom table: drugs), where the compounds (rows) are grouped by their respective target proteins. An additional group is created for the user-input compounds since their target information is not presented. This information is shown under the "Target Information" column. iBioProVis provides UniProt protein accessions, gene names and ChEMBL identifiers for the target proteins. In addition to these, compound ChEMBL ID, molecular formulas and PubChem cross references are given under this table, for the selected compounds. The second (bottom) table is reserved to present only the approved or experimental drugs in the user selection. Here, iBioProVis provides drug names, and clinical trial cross references in addition to the aforementioned information. At the top right side of the plot, there are buttons for easy navigation on the plot such as pan, box zoom, box select, wheel zoom, tap, reset and save. There is a bioactivity value filter at the bottom of the plot, which can be used interactively to remove the compounds that do not satisfy the selected bioactivity threshold (against the corresponding target protein(s)), from the plot.

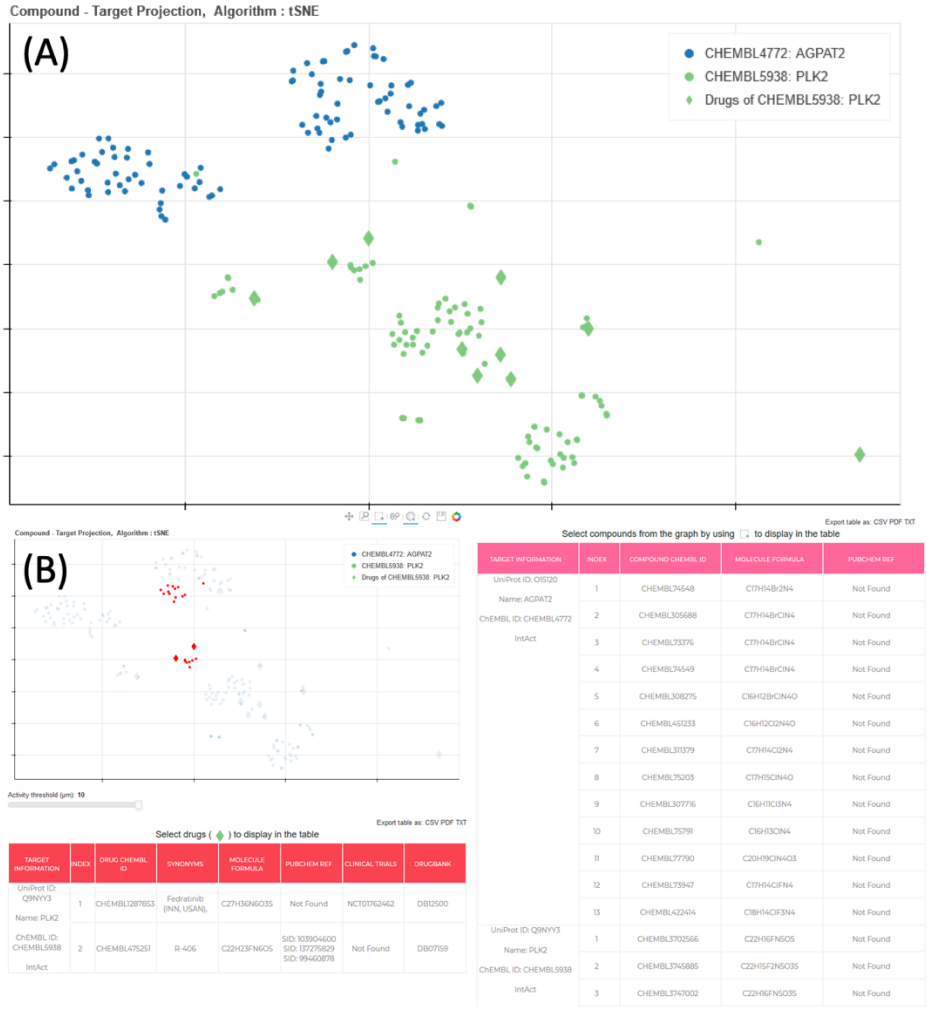


Figure 6.1. Interactive output of the iBioProVis framework. (A) Embedding output without user selection. (B) The same embedding output (including tables) after user selection of compounds. Visual clusters and the nodes in close vicinity may indicate the same or similar target proteins.

6.3 Discussion and Conclusion

iBioProVis is an unprecedented framework that can be utilized for virtual screening and for chemical genomics. It can be used for several purposes, including the investigation and analysis of how active compounds of different target proteins are distributed on a 2-D space, as well as the prediction of bioactivity profiles for new

or uncharacterized compounds, based on the features of compounds with known bioactivity information. Furthermore, it may provide insight to drug repurposing studies by identifying the compounds that are embedded close to an approved drug, especially when those compounds are known binders of a different target protein.

CHAPTER 7

DISCUSSION & CONCLUSION

In this thesis, several deep learning solutions were proposed for protein function prediction and drug-target interaction problems. In Chapter 2, our deep learning-based protein function prediction method, DEEPred, was explained. DEEPred uses stacked feed-forward deep neural networks by incorporating GO hierarchy to provide reliable prediction. The performance of DEEPred was compared with the state-of-the art methods and it is showed that DEEPred performs better than the existing methods in general. In addition, new models were trained with DEEPred for CAFA PI challenge and DEEPred ranked fourth in one of the three categories in CAFA PI. In Chapter 3, DeepScreen method is described which is a drug-target interaction prediction that employs convolutional neural networks which takes compound images as input to the neural network with the aim of extracting compound features automatically from the input images. Benchmarking results of DeepScreen were compared with state-of-the art on multiple datasets and *in vitro* experimental validations were also performed for selected predictions in Cancer Systems Laboratory. In Chapter 4, MDeePred method was explained for binding affinity prediction which uses pairwise input neural networks along with the proposed multi-channel protein input representations. The performance of MDeePred was compared with recently proposed binding affinity prediction methods using the same experimental settings. In Chapter 5, iBioProVis tool was explained which is a visualization tool for chemical space.

Although the proposed methods achieved better predictive performances than the state-of-the art methods in majority of the cases, there are still room for significant improvements and limitations in the field. Below, my observations are first discussed along with the current limitations and possible solutions. Finally, the perspectives and future work are explained.

7.1 Observations, Limitations and Solutions

One of the most important steps in drug-target interaction prediction method development is the training dataset construction. Therefore, as in all machine learning applications, training sets should contain reliably labelled data. For example, compounds should be labelled as active and inactive against target proteins based on predefined thresholds. Although IC50 values of 10 μM/1 μM (or lower) values are widely used in the literature, selection of the activity threshold is a problem specific issue and it is not clear what should be the activity threshold value to assume positive interaction.

Even though there are millions of entries in the compound and bioactivity databases, the quality of available data is an important issue. For example, in PubChem database the entries mostly consist of the deposited data by different providers and majority of the drug-target interactions are just reported as active or inactive without providing binding affinity values. In addition, the experimental conditions of the experiments are not known in majority of the data points. Therefore, it is not possible to infer the real bioactivity values for these entries. In order to overcome this issue; well-defined, solid protocols should be introduced to publish the outcomes of experimental results so that the noise will be reduced and accurate annotations can be done.

Another limitation regarding dataset construction is the difference between the number of active and inactive compounds after the activity thresholds are applied which is known as *class imbalance* problem in machine learning. In compound and bioactivity databases such as ChEMBL, majority of the bioactivity values correspond to lower bioactivity values (i.e., less than 10 μM). Therefore, when a threshold such as 10 μM is applied, it is seen that in several cases number of negative samples is less than number of positive samples which means that negative data point scarcity is more pronounced in negative training dataset selection. Since low IC50 values are generally desirable, experimentally observed high activity values are often not reported in the literature and in the bioactivity databases. In the end, there very

few instances to be used as negative training instances. One of the widely accepted solutions to this problem is selecting random data points (after removing positive data points) from all possible drug-target pairs with the assumption that the ratio of truly active to inactive pairs is so low that random selection would yield a good quality negatives set. However, this is not always guaranteed as knowledge regarding the truly active to inactive ratio is not known. There are also alternative solutions to this problem such as the advanced sampling techniques [312].

Although training of very deep and complex networks become possible with the advancement of technology, training of these networks requires to adjust extensive hyperparameter set which takes significant amount of time. Therefore, one of the most important challenges regarding the training of deep learning-based methods is still the *computational complexity*. Although there is a growing field of research on new algorithmic approaches to reduce the complexity of DNN-based techniques without compromising from the prediction performance and more advanced GPU/TPU hardware technologies are being developed, there is still some time before these systems (in terms of both hardware and software) to become easily affordable.

Most of the deep learning-based studies so far emphasized the potential and applicability of DNNs for the development of efficient virtual screening methods; however, there are no public production pipelines to predict and publish large-scale DTIs. Considering the current availability of the chemical structures and bioactivity information in public databases, which is required for constructing such pipelines, we expect to see DNN-based large scale analyses and novel web-services presenting their results in the near future.

7.2 Perspectives and Future Directions

Deep learning techniques have shown significantly better performance for DTI prediction compared to the conventional machine learning methods. In conventional methods, the problem is divided into different parts and each part is solved

individually. Whereas the main advantage deep learning algorithms is automatic feature extraction from raw data. Therefore, it is more suitable for end-to-end learning. As a result, we expect a significant shift, not only in virtual screening but also in the drug discovery field in general, towards utilizing novel deep learning based architectures in the near future. Besides, the flexibility of deep learning architectures allows researchers to model drug-target interactions in various ways, each of which may have specific advantages.

Recently, *de novo* drug design using deep generative neural networks is becoming more popular in computational drug discovery where the aim is to generate compounds with desired properties based on the previous experimental results. We expect that, with the employment of DNN based models, the field of *de novo* drug design will start to produce truly novel drug candidates in the near future. One of the advantages of generating drug-like compounds is that generated compounds can be searched and purchased from the databases such as MolPort database.

Deep learning applications have shown significant improvements compared to traditional machine learning methods in recent years. However, interpretability of the deep learning models and their predictions still remains as the key limitation of deep learning studies. Several studies have been proposed in order to overcome interpretability issue in deep learning methods [313], [314]. For example, Grad-Cam is a method that produce visual explanations of predictions by analyzing the change of gradients and neuron activations [315]. Although there are a few studies in the drug discovery literature that proposed solutions for interpretability, this area should be investigated thoroughly which could be a future direction in computational drug discovery area [316]–[318].

We expect that integration of large-scale omic data (e.g., transcriptomics, interactomics, epigenomics, metabolomics, functional genomics, etc.) at the input level will become popular in the near future to increase both the quality and the coverage of DTI predictions. Conventionally, known bioactivities are used along with the structural attributes of compounds and/or target proteins to model the drug-

target interactions. However, the recent accumulation of the omic data presents opportunities for the identification of the unknown parts of the drug-target interaction space. The expected contribution of the omic approach mainly comes from integrating different types of features in an ensemble/hybrid setting, where different features complement each other to produce a more complete picture. Since components of the omic data have different structures (e.g., interactomic data mostly define the pairwise relations between proteins, transcriptomic data displays quantitative measurements in terms of how the expression of genes change under different conditions), generation of the feature vectors with the standardization of the information have critical importance.

Majority of the targets have low number of bioactivity data associated with them and it is not possible to train robust models using conventional classifiers for these targets due to the low number of training samples. Recently, one-shot and low-shot-learning methods have been proposed to overcome this issue and these methods can provide predictions even for the classes with a few samples [135], [319]–[321]. More studies are expected to be published on this issue.

The Anatomical Therapeutic Chemical (ATC) Classification System provides valuable information for the classification of drugs in terms of their therapeutic effect, and their pharmacological and physicochemical properties. Assigning an ATC code to a compound requires curation efforts, as a result, only approved and experimental drugs have ATC code annotations. Large-scale prediction of ATC codes for all compounds recorded in chemical databases can help to identify the roles for these compounds. In addition, predicting new ATC codes for known drugs can be used to aid drug repositioning. Currently, there are only a few ATC code prediction studies in the literature, most of which have been proposed in the last few years [14], [322], [323]. It is expected to see more studies of this kind in the future.

Researchers are now focusing on incorporating different entities (e.g., drugs, targets, pathways, diseases etc.) into large knowledge graphs and perform predictions by running algorithms to predict different relations among entities. Recently, several

deep learning algorithms have been proposed recently to be run on large knowledge graphs [233], [324]. I believe that this will be one of the major research directions in the near future.

Recently, there are a few examples of in-silico generated drug candidates have been shown to pass first stages of drug discovery pipeline. I believe that in the near future, more of these works will be conducted and artificial intelligence and machine learning algorithms will play an increasingly important role in drug discovery. It is expected that novel compounds will be designed, synthesized and tested automatically and more accurate drug candidates will be produced with the combined work of experts and machine learning algorithms.

Finally, as mentioned earlier, although several algorithms have been proposed, these methods are mostly developed to show the applicability and potential of computational methods in drug discovery field. Therefore, they are far from to be integrated in a real drug discovery and development pipeline. As a future work, our aim is to collaborate with pharmaceutical researchers and companies to better understand their needs and integrate our methods to a real drug discovery pipeline.

REFERENCES

[1] S. Kim *et al.*, "PubChem Substance and Compound databases.," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1202-13, 2016.

[2] A. P. Bento *et al.*, "The ChEMBL bioactivity database: An update," *Nucleic Acids Res.*, vol. 42, no. D1, pp. 1083–1090, 2014.

[3] A. Williams and V. Tkachenko, "The Royal Society of Chemistry and the delivery of chemistry data repositories for the community," *J. Comput. Aided. Mol. Des.*, pp. 1023–1030, 2014.

[4] J. Hastings *et al.*, "The ChEBI reference database and ontology for biologically relevant chemistry: Enhancements for 2013," *Nucleic Acids Res.*, vol. 41, no. D1, pp. 456–463, 2013.

[5] V. Law *et al.*, "DrugBank 4.0: Shedding new light on drug metabolism," *Nucleic Acids Res.*, vol. 42, no. D1, pp. 1091–1097, 2014.

[6] T. U. Consortium, "UniProt: a hub for protein information," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D204–D212, 2014.

[7] A. S. Rifaioglu, H. Atas, M. J. Martin, R. Cetin-Atalay, V. Atalay, and T. Doğan, "Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases," *Brief. Bioinform.*, vol. 20, no. 5, pp. 1878–1912, 2019.

[8] R. P. Hertzberg and A. J. Pope, "High-throughput screening: New technology for the 21st century," *Curr. Opin. Chem. Biol.*, vol. 4, no. 4, pp. 445–451, 2000.

[9] A. L. Hopkins, "Drug discovery: Predicting promiscuity," *Nature*, vol. 462, no. 7270, pp. 167–168, 2009.

[10] S. M. Paul *et al.*, "How to improve R&D productivity: the pharmaceutical

industry's grand challenge.," *Nat. Rev. Drug Discov.*, vol. 9, no. 3, pp. 203–14, 2010.

[11] A. Bateman *et al.*, "UniProt: A worldwide hub of protein knowledge," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D506–D515, 2019.

[12] A. C. Schierz, "Virtual screening of bioassay data," *J. Cheminform.*, vol. 1, no. 1, pp. 1–12, 2009.

[13] H. Iwata, R. Sawada, S. Mizutani, M. Kotera, and Y. Yamanishi, "Large-Scale Prediction of Beneficial Drug Combinations Using Drug Efficacy and Target Profiles," *J. Chem. Inf. Model.*, vol. 55, pp. 2705–2716, 2015.

[14] Z. Liu *et al.*, "Similarity-based prediction for Anatomical Therapeutic Chemical classification of drugs by integrating multiple data sources," *Bioinformatics*, vol. 31, no. 11, pp. 1788–1795, 2015.

[15] L. Chen, W. M. Zeng, Y. D. Cai, K. Y. Feng, and K. C. Chou, "Predicting anatomical therapeutic chemical (ATC) classification of drugs by integrating chemical-chemical interactions and similarities," *PLoS One*, vol. 7, no. 4, pp. 1–7, 2012.

[16] T. Ashburn and K. B. Thor, "Drug repositioning: identifying and developing new uses for existing drugs," *Nat. Rev. Drug Discov.*, vol. 3, pp. 673–683, 2004.

[17] C. CR and D. Sullivan, "New uses for old drugs.," *Infect. Dis. Clin. North Am.*, vol. 3, no. 3, pp. 653–664, 1989.

[18] M. S. Boguski, K. D. Mandl, and V. P. Sukhatme, "Drug discovery. Repurposing with a difference.," *Science*, vol. 324, no. 5933, pp. 1394–1395, 2009.

[19] B. K. Shoichet, "Virtual screening of chemical libraries.," *Nature*, vol. 432, no. 7019, pp. 862–865, 2004.

[20] J. Singh *et al.*, “Successful shape-based virtual screening: the discovery of a potent inhibitor of the type I TGFbeta receptor kinase (TbetaRI).,” *Bioorg. Med. Chem. Lett.*, vol. 13, no. 24, pp. 4355–4359, 2003.

[21] O. M. Becker *et al.*, “An integrated in silico 3D model-driven discovery of a novel, potent, and selective amidosulfonamide 5-HT1A agonist (PRX-00023) for the treatment of anxiety and depression,” *J. Med. Chem.*, vol. 49, no. 11, pp. 3116–3135, 2006.

[22] R. C. Rizzo, D.-P. Wang, J. Tirado-rives, and W. L. Jorgensen, “Validation of a Model for the Complex of HIV-1 Reverse Transcriptase with Sustiva through Computation of Resistance Profiles,” *J Am Chem Soc*, vol. 122, no. 11, pp. 12898–12900, 2000.

[23] A. Brik *et al.*, “Rapid Diversity-Oriented Synthesis in Microtiter Plates for in Situ Screening of HIV Protease Inhibitors,” *ChemBioChem*, vol. 4, no. 11, pp. 1246–1248, 2003.

[24] M. J. Keiser *et al.*, “Predicting new molecular targets for known drugs.,” *Nature*, vol. 462, no. 7270, pp. 175–181, 2009.

[25] T. N. Doman *et al.*, “Molecular docking and high-throughput screening for novel inhibitors of protein tyrosine phosphatase-1B,” *J. Med. Chem.*, vol. 45, no. 11, pp. 2213–2221, 2002.

[26] R. A. Powers, F. Morandi, and B. K. Shoichet, “Structure-based discovery of a novel, noncovalent inhibitor of AmpC β-lactamase,” *Structure*, vol. 10, no. 7, pp. 1013–1023, 2002.

[27] M. A. Yıldırım, K.-I. Goh, M. E. Cusick, A.-L. Barabási, and M. Vidal, “Drug—target network,” *Nat. Biotechnol.*, vol. 25, no. 10, pp. 1119–1126, 2007.

[28] A. L. Hopkins, “Network pharmacology: the next paradigm in drug discovery,” *Nat. Chem. Biol.*, vol. 4, pp. 682–690, 2008.

[29] H. Li *et al.*, "TarFisDock: A web server for identifying drug targets with docking approach," *Nucleic Acids Res.*, vol. 34, no. WEB. SERV. ISS., pp. 219–224, 2006.

[30] S. L. Kinnings and R. M. Jackson, "ReverseScreen3D : A Structure-Based Ligand Matching Method To Identify Protein Targets," *J. Chem. Inf. Model.*, pp. 624–634, 2011.

[31] P. Ripphausen, B. Nisius, and J. Bajorath, "State-of-the-art in ligand-based virtual screening," *Drug Discov. Today*, vol. 16, no. 9–10, pp. 372–376, 2011.

[32] B. Chen *et al.*, "Evaluation of machine-learning methods for ligand-based virtual screening," *J. Comput. Aided. Mol. Des.*, vol. 21, no. 1–3, pp. 53–62, 2007.

[33] T. Cheng, Q. Li, Z. Zhou, Y. Wang, and S. H. Bryant, "Structure-based virtual screening for drug discovery: a problem-centric review.," *AAPS J.*, vol. 14, no. 1, pp. 133–41, Mar. 2012.

[34] E. Lionta, G. Spyrou, D. K. Vassilatis, and Z. Cournia, "Structure-based virtual screening for drug discovery: principles, applications and recent advances.," *Curr. Top. Med. Chem.*, vol. 14, no. 16, pp. 1923–38, 2014.

[35] H. Geppert, M. Vogt, and J. Bajorath, "Current Trends in Ligand-Based Virtual Screening: Molecular Representations, Data Mining Methods, New Application Areas, and Performance Evaluation," *J. Chem. Inf. Model.*, vol. 50, no. 2, pp. 205–216, Feb. 2010.

[36] A. Lavecchia and C. Di Giovanni, "Virtual screening strategies in drug discovery: a critical review.," *Curr. Med. Chem.*, vol. 20, no. 23, pp. 2839–60, 2013.

[37] G. J. P. van Westen, J. K. Wegner, A. P. IJzerman, H. W. T. van Vlijmen, and A. Bender, "Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets," *Med. Chem. Commun.*,

vol. 2, no. 1, pp. 16–30, 2011.

[38] G. J. P. Van Westen *et al.*, "Benchmarking of protein descriptor sets in proteochemometric modeling (part 1): comparative study of 13 amino acid descriptor sets," *J. Cheminform.*, vol. 5, no. 9, pp. 1–11, 2013.

[39] T. Qiu *et al.*, "The recent progress in proteochemometric modelling: focusing on target descriptors, cross-term descriptors and application scope," *Brief. Bioinform.*, vol. 18, no. 1, pp. 125–136, Jan. 2017.

[40] I. Cortés-Ciriano *et al.*, "Polypharmacology modelling using proteochemometrics (PCM): recent methodological developments, applications to target families, and future prospects," *Med. Chem. Commun.*, vol. 6, no. 1, 2015.

[41] M. Zheng, X. Liu, Y. Xu, H. Li, C. Luo, and H. Jiang, "Computational methods for drug design and discovery: Focus on China," *Trends Pharmacol. Sci.*, vol. 34, no. 10, pp. 549–559, 2013.

[42] A. Koutsoukas *et al.*, "From in silico target prediction to multi-target drug design : Current databases , methods and applications," *J. Proteomics*, vol. 74, no. 12, pp. 2554–2574, 2011.

[43] E. Glaab, "Building a virtual ligand screening pipeline using free software: A survey," *Brief. Bioinform.*, vol. 17, no. 2, pp. 352–366, 2016.

[44] G. M. Morris *et al.*, "AutoDock-related material Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function," *Comput. Chem. J. Comput. Chem*, vol. 19, no. 28, pp. 1639–1662, 1998.

[45] T. J. Ewing, S. Makino, A. G. Skillman, and I. D. Kuntz, "DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases.," *J. Comput. Aided. Mol. Des.*, vol. 15, no. 5, pp. 411–28, May 2001.

[46] R. A. Friesner *et al.*, "Glide: A New Approach for Rapid, Accurate Docking

and Scoring. 1. Method and Assessment of Docking Accuracy," *J. Med. Chem.*, vol. 47, no. 7, pp. 1739–1749, 2004.

[47] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, and R. D. Taylor, "Improved Protein – Ligand Docking Using GOLD," *Proteins Struct. Funct. Bioinforma.*, vol. 623, no. January, pp. 609–623, 2003.

[48] B. Kramer, M. Rarey, and T. Lengauer, "Evaluation of the FlexX incremental construction algorithm for protein- ligand docking," *Proteins Struct. Funct. Genet.*, vol. 37, no. 2, pp. 228–241, 1999.

[49] M. McGann, "FRED Pose Prediction and Virtual Screening Accuracy," *J. Chem. Inf. Model.*, vol. 51, no. 3, pp. 578–596, Mar. 2011.

[50] J. Marialke, S. Tietze, and J. Apostolakis, "Similarity Based Docking," *J. Chem. Inf. Model.*, vol. 48, no. 1, pp. 186–196, Jan. 2008.

[51] M. Brylinski, "Nonlinear scoring functions for similarity-based ligand docking and binding affinity prediction," *J. Chem. Inf. Model.*, vol. 53, no. 11, pp. 3097–3112, 2013.

[52] T. Kawabata and H. Nakamura, "3D flexible alignment using 2D maximum common substructure: Dependence of prediction accuracy on target-reference chemical similarity," *J. Chem. Inf. Model.*, vol. 54, no. 7, pp. 1850–1863, 2014.

[53] M. N. Drwal and R. Griffith, "Combination of ligand- and structure-based methods in virtual screening," *Drug Discov. Today Technol.*, vol. 10, no. 3, pp. e395–e401, 2013.

[54] G. Sliwoski, S. Kothiwale, J. Meiler, and E. W. Lowe, "Computational Methods in Drug Discovery," *Pharmacol. Rev.*, vol. 66, no. January, pp. 334–395, 2014.

[55] E. Lounkine *et al.*, "Large-scale prediction and testing of drug activity on side-effect targets.," *Nature*, vol. 486, no. 7403, pp. 361–7, 2012.

[56] R. Sawada, H. Iwata, S. Mizutani, and Y. Yamanishi, “Target-Based Drug Repositioning Using Large-Scale Chemical-Protein Interactome Data,” *J. Chem. Inf. Model.*, vol. 55, no. 12, pp. 2717–2730, 2015.

[57] D. M. Krüger and A. Evers, “Comparison of structure- and ligand-based virtual screening protocols considering hit list complementarity and enrichment factors,” *ChemMedChem*, vol. 5, no. 1, pp. 148–158, 2010.

[58] T. J. Marrone, J. M. Briggs, and J. a McCammon, “Structure-based drug design: computational advances.,” *Annu. Rev. Pharmacol. Toxicol.*, vol. 37, pp. 71–90, 1997.

[59] A. C. Anderson, “The Process of Structure-Based Drug Design,” *Cell Chem. Biol.*, vol. 128, no. 2, pp. 189–190, 2014.

[60] C. M. Bishop, *Pattern recognition and machine learning. (Information Science and Statistics)*. Springer, 2006.

[61] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.

[62] D. Xu and Y. Tian, “A Comprehensive Survey of Clustering Algorithms,” *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, 2015.

[63] J. Pérez-sianes, H. Pérez-sánchez, and F. Díaz, “Virtual Screening : A Challenge for Deep Learning,” *10th Int. Conf. PACBB, Adv. Intell. Syst. Comput.*, pp. 13–22, 2016.

[64] J. D. MacCuish and N. E. MacCuish, *Clustering in Bioinformatics and Drug Discovery*. 2011.

[65] G. Drakakaki *et al.*, “Clusters of bioactive compounds target dynamic endomembrane networks in vivo,” *PNAS*, vol. 108, no. 43, pp. 17850–17855, 2011.

[66] P. Larrañaga *et al.*, “Machine learning in bioinformatics,” *Brief. Bioinform.*,

vol. 7, no. 1, pp. 86–112, 2006.

[67] M. W. Libbrecht and W. S. Noble, “Machine learning applications in genetics and genomics,” *Nat Rev Genet*, vol. 16, no. 6, pp. 321–332, 2015.

[68] L. J. Jensen and A. Bateman, “The rise and fall of supervised machine learning techniques,” *Bioinformatics*, vol. 27, no. 24, pp. 3331–3332, 2011.

[69] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, “Applications of Deep Learning in Biomedicine,” *Mol. Pharm.*, vol. 13, no. 5, pp. 1445–1454, 2016.

[70] D. Butina, “Unsupervised Data Base Clustering Based on Daylight’s Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets,” *J. Chem. Inf. Model.*, vol. 39, pp. 747–750, 1999.

[71] Bisgin *et al.*, “Mining FDA drug labels using an unsupervised learning technique - topic modeling,” *BMC Bioinformatics*, vol. 12, no. suppl10, p. S11 (8 pp.), 2011.

[72] J. Hert, M. J. Keiser, J. J. Irwin, T. I. Oprea, and B. K. Shoichet, “Quantifying the relationships among drug classes,” *J. Chem. Inf. Model.*, vol. 48, no. 4, pp. 755–765, 2008.

[73] N. J. Perualila-Tan, Z. Shkedy, W. Talloen, H. W. H. Göhlmann, M. Van Moerbeke, and A. Kasim, “Weighted similarity-based clustering of chemical structures and bioactivity data in early drug discovery,” *J. Bioinform. Comput. Biol.*, vol. 14, no. 4, p. 1650018, 2016.

[74] S. Korkmaz, G. Zararsiz, and D. Goksuluk, “MLViS: A Web Tool for Machine Learning- Based Virtual Screening in Early-Phase of Drug Discovery and Development,” *PLoS One*, vol. 10, no. 4, pp. 1–15, 2015.

[75] H. Ding, H. Mamitsuka, and S. Zhu, “Similarity-based machine learning methods for predicting drug-target interactions : a brief review,” *Brief. Bioinform.*, vol. 15, no. 5, pp. 734–747, 2013.

[76] D. Gfeller, A. Grosdidier, M. Wirth, A. Daina, O. Michielin, and V. Zoete, “SwissTargetPrediction: A web server for target prediction of bioactive small molecules,” *Nucleic Acids Res.*, vol. 42, no. W1, pp. 32–38, 2014.

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

[78] H. Lim, P. Gray, L. Xie, and A. Poleksic, “Improved genome-scale multi-target virtual screening via a novel collaborative filtering approach to cold-start problem,” *Sci. Rep.*, vol. 6, no. 1, p. 38860, Dec. 2016.

[79] Y. Yamanishi, 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, pp. 232–240, 2008.

[80] G. Cano *et al.*, “Automatic selection of molecular descriptors using random forest: Application to drug discovery,” *Expert Syst. Appl.*, vol. 72, pp. 151–159, Apr. 2017.

[81] H. Yabuuchi *et al.*, “Analysis of multiple compound–protein interactions reveals novel bioactive molecules,” *Mol. Syst. Biol.*, vol. 7, no. 5, pp. 1–12, 2011.

[82] Y. Okuno *et al.*, “GLIDA: GPCR - Ligand database for chemical genomics drug discovery - Database and tools update,” *Nucleic Acids Res.*, vol. 36, no. SUPPL. 1, pp. 907–912, 2008.

[83] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, “KEGG as a reference resource for gene and protein annotation,” *Nucleic Acids Res.*, vol. 44, no. D1, pp. D457–D462, 2016.

[84] “OrangeBook,” *https://www.fda.gov/Drugs/InformationOnDrugs/ucm129662.htm, Accessed on January 8, 2018*. [Online]. Available:

https://www.accessdata.fda.gov/scripts/cder/ob/. [Accessed: 17-Oct-2017].

[85] Z. Li *et al.*, "In silico prediction of drug-target interaction networks based on drug chemical structure and protein sequences," *Sci. Rep.*, vol. 7, no. 1, p. 11174, 2017.

[86] Y. Yamanishi, M. Kotera, M. Kanehisa, and S. Goto, "Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework," *Bioinformatics*, vol. 26, no. 12, pp. 246–254, 2010.

[87] M. Gönen, "Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization," *Bioinformatics*, vol. 28, no. 18, pp. 2304–2310, 2012.

[88] X. Chen, C. C. Yan, X. Zhang, X. Zhang, and F. Dai, "Drug – target interaction prediction : databases , web servers and computational models," *Brief. Bioinform.*, vol. 17, no. 4, pp. 696–712, 2016.

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

[90] D. Rognan, "Chemogenomic approaches to rational drug design," *Br. J. Pharmacol.*, vol. 152, no. 1, pp. 38–52, 2007.

[91] L. Jacob and J. Vert, "Protein – ligand interaction prediction : An improved chemogenomics approach," *Bioinformatics*, vol. 24, no. 19, pp. 2149–2156, 2008.

[92] R. Sawada, M. Kotera, and Y. Yamanishi, "Benchmarking a wide range of chemical descriptors for drug-target interaction prediction using a chemogenomic approach," *Mol. Inform.*, vol. 33, no. 11–12, pp. 719–731, 2014.

[93] W. Ba-alawi, O. Soufan, M. Essack, P. Kalnis, and V. B. Bajic, "DASPfind: new efficient method to predict drug–target interactions," *J. Cheminform.*,

vol. 8, no. 1, p. 15, Dec. 2016.

[94] R. S. Olayan, H. Ashoor, and V. B. Bajic, “DDR: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches,” *Bioinformatics*, vol. 34, no. 7, pp. 1164–1173, Apr. 2018.

[95] H. Liu, J. Sun, J. Guan, J. Zheng, and S. Zhou, “Improving compound-protein interaction prediction by building up highly credible negative samples,” *Bioinformatics*, vol. 31, no. 12, pp. i221–i229, 2015.

[96] Y. Park and E. M. Marcotte, “Revisiting the negative example sampling problem for predicting protein-protein interactions,” *Bioinformatics*, vol. 27, no. 21, pp. 3024–3028, 2011.

[97] A. Ben-Hur and W. S. Noble, “Choosing negative examples for the prediction of protein-protein interactions.,” *BMC Bioinformatics*, vol. 7 Suppl 1, p. S2, 2006.

[98] H. Iwata, R. Sawada, S. Mizutani, and Y. Yamanishi, “Systematic Drug Repositioning for a Wide Range of Diseases with Integrative Analyses of Phenotypic and Molecular Data,” *J. Chem. Inf. Model.*, vol. 55, p. 446−459, 2015.

[99] C. Nantasenamat, C. Isarankura-Na-Ayudhya, T. Naenna, and V. Prachayasittikul, “A practical overview of quantitative structure-activity relationship,” *EXCLI J.*, vol. 8, pp. 74–88, 2009.

[100] R. Kurczab, S. Smusz, and A. J. Bojarski, “The influence of negative training set size on machine learning-based virtual screening,” *J. Cheminform.*, vol. 6, no. 1, p. 32, Dec. 2014.

[101] O. Soufan, W. Ba-Alawi, M. Afeef, M. Essack, P. Kalnis, and V. B. Bajic, “DRABAL: novel method to mine large high-throughput screening assays using Bayesian active learning,” *J. Cheminform.*, vol. 8, no. 1, pp. 1–14, 2016.

[102] O. Soufan *et al.*, "Mining Chemical Activity Status from High-Throughput Screening Assays," *PLoS One*, vol. 10, no. 12, pp. 1–16, 2015.

[103] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[104] W. B. Powell and Wiley InterScience (Online service), "Approximate dynamic programming : solving the curses of dimensionality," *Wiley Intersci.*, p. 627, 2011.

[105] M. Hall, "Correlation-based Feature Selection for Machine Learning," *PhD thesis*, vol. 21i195-i20, no. April, pp. 1–5, 1999.

[106] D. L. Padmaja and B. Vishnuvardhan, "Comparative Study of Feature Subset Selection Methods for Dimensionality Reduction on Scientific Data," *2016 IEEE 6th Int. Conf. Adv. Comput.*, pp. 31–34, 2016.

[107] A. Janecek, W. N. W. Gansterer, M. Demel, and G. Ecker, "On the Relationship Between Feature Selection and Classification Accuracy.," *Fsdm*, vol. 4, pp. 90–105, 2008.

[108] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[109] Y. Liu, "A comparative study on feature selection methods for drug discovery," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 5, pp. 1823–1828, 2004.

[110] I. K. Fodor, "A survey of dimension reduction techniques," *Library (Lond).*, vol. 18, no. 1, pp. 1–18, 2002.

[111] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Process. Mag.*, no. November, pp. 82–97, 2012.

[112] L. Deng, G. Hinton, and B. Kingsbury, "New Types of Deep Neural Network Learning For Speech Recognition And Related Applications : An Overview." pp. 1–5, 2013.

[113] C. Angermueller, T. Pärnamaa, L. Parts, O. Stegle, and S. Oliver, “Deep Learning for Computational Biology,” *Mol. Syst. Biol.*, vol. 12, no. 878, pp. 1–16, 2016.

[114] S. Min, B. Lee, and S. Yoon, “Deep learning in bioinformatics,” *Brief. Bioinform.*, vol. 18, no. 5, pp. 851–869, 2016.

[115] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[116] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, “Multi-task Neural Networks for QSAR Predictions,” *arXiv*, pp. 1–21, 2014.

[117] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, “Deep Neural Nets as a Method for Quantitative Structure − Activity Relationships,” *J. Chem. Inf. Model.*, vol. 55, p. 263−274, 2015.

[118] E. Gawehn, J. A. Hiss, and G. Schneider, “Deep Learning in Drug Discovery,” *Mol. Inform.*, vol. 35, pp. 3–14, 2016.

[119] I. I. Baskin, D. Winkler, and I. V Tetko, “A renaissance of neural networks in drug discovery,” *Expert Opin. Drug Discov. ISSN*, vol. 11, no. 8, pp. 785–795, 2016.

[120] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, “DeepTox : Toxicity Prediction using Deep Learning,” *Front. Environ. Sci.*, vol. 3, no. 80, pp. 1–15, 2016.

[121] B. Ramsundar *et al.*, “Massively Multitask Networks for Drug Discovery,” *arXiv*, pp. 1–27, 2015.

[122] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, “Deep neural nets as a method for quantitative structure-activity relationships,” *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, 2015.

[123] I. Wallach, M. Dzamba, and A. Heifets, “AtomNet : A Deep Convolutional

Neural Network for Bioactivity Prediction in Structure-based Drug Discovery," *arXiv*, vol. arXiv:1510, pp. 1–11, 2015.

[124] T. Unterthiner, A. Mayr, G. Klambauer, M. Steijaert, J. K. Wegner, and H. Ceulemans, "Deep Learning as an Opportunity in Virtual Screening," *Deep Learn. Represent. Learn. Work. NIPS 2014*, pp. 1–9, 2014.

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

[126] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, "Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data," *J. Cheminform.*, vol. 9, no. 1, pp. 1–13, 2017.

[127] F. Wan and J. Zeng, "Deep learning with feature embedding for compound-protein interaction prediction," *bioRxiv*, pp. 0–20, Nov. 2016.

[128] E. B. Lenselink *et al.*, "Beyond the hype: Deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set John Holliday," *J. Cheminform.*, vol. 9, no. 1, pp. 1–14, 2017.

[129] G. B. Goh, N. O. Hodas, C. Siegel, and A. Vishnu, "SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties," *arXiv*, pp. 1–8, 2017.

[130] M. Wen *et al.*, "Deep-Learning-Based Drug-Target Interaction Prediction," *J. Proteome Res.*, vol. 16, no. 4, pp. 1401–1409, 2017.

[131] Y. Wang and J. Zeng, "Predicting drug-target interactions using restricted Boltzmann machines," in *Bioinformatics*, 2013, vol. 29, no. 13, pp. i126–i134.

[132] A. Gonczarek, J. M. Tomczak, S. Zareba, J. Kaczmar, P. Dabrowski, and M. J. Walczak, "Interaction prediction in structure-based virtual screening using

deep learning," *Comput. Biol. Med.*, pp. 1–6, 2017.

[133] G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas, and N. Baker, "Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models," *PNNL pub*, pp. 1–38, 2017.

[134] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *J. Comput. Aided. Mol. Des.*, vol. 30, no. 8, pp. 595–608, 2016.

[135] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low Data Drug Discovery with One-Shot Learning," *ACS Cent. Sci.*, vol. 3, no. 4, pp. 283–293, 2017.

[136] J. Desaphy, G. Bret, D. Rognan, and E. Kellenberger, "Sc-PDB: A 3D-database of ligandable binding sites-10 years on," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D399–D404, 2015.

[137] M. Kuhn, I. Letunic, L. J. Jensen, and P. Bork, "The SIDER database of drugs and side effects.," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1075-9, 2016.

[138] S. G. Rohrer and K. Baumann, "Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data," *J. Chem. Inf. Model.*, vol. 49, no. 2, pp. 169–184, Feb. 2009.

[139] M. I. Davis *et al.*, "Comprehensive analysis of kinase inhibitor selectivity," *Nat. Biotechnol.*, vol. 29, no. 11, pp. 1046–1051, 2011.

[140] J. Tang *et al.*, "Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis," *J. Chem. Inf. Model.*, vol. 54, no. 3, pp. 735–743, 2014.

[141] D. S. Wishart *et al.*, "DrugBank 5.0: A major update to the DrugBank database for 2018," *Nucleic Acids Res.*, vol. 46, no. D1, pp. D1074–D1082, 2018.

[142] C. Southan *et al.*, "The IUPHAR/BPS Guide to PHARMACOLOGY in 2016: Towards curated quantitative interactions between 1300 protein targets and 6000 ligands," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1054–D1068, 2016.

[143] V. G. Maltarollo, J. C. Gertrudes, P. R. Oliveira, and K. M. Honorio, "Applying machine learning techniques for ADME-Tox prediction: a review," *Expert Opin. Drug Metab. Toxicol.*, vol. 11, no. 2, pp. 259–271, Feb. 2015.

[144] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov, "Deep Learning Applications for Predicting Pharmacological Properties of Drugs and Drug Repurposing Using Transcriptomic Data.," *Mol. Pharm.*, vol. 13, no. 7, pp. 2524–30, Jul. 2016.

[145] A. Lusci, G. Pollastri, and P. Baldi, "Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules.," *J. Chem. Inf. Model.*, vol. 53, no. 7, pp. 1563–75, Jul. 2013.

[146] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges.," *Brief. Bioinform.*, pp. 1–11, May 2017.

[147] T. Ching *et al.*, "Opportunities And Obstacles For Deep Learning In Biology And Medicine," *bioRxiv*, p. 142760, 2017.

[148] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," *J. Comput. Chem.*, vol. 38, no. 16, pp. 1291–1307, 2017.

[149] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PLoS One*, vol. 12, no. 6, pp. 1–17, 2017.

[150] M. Kuhn *et al.*, "STITCH 4: Integration of protein-chemical interactions with user data," *Nucleic Acids Res.*, vol. 42, no. D1, pp. 401–407, 2014.

[151] M. K. Gilson, T. Liu, M. Baitaluk, G. Nicola, L. Hwang, and J. Chong,

"BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1045–D1053, 2016.

[152] A. Ahmed, R. D. Smith, J. J. Clark, J. B. D. Jr, and H. A. Carlson, "Recent improvements to Binding MOAD: A resource for protein-ligand Binding affinities and structures," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D465–D469, 2015.

[153] Y. Liu, Q. Wei, G. Yu, W. Gai, Y. Li, and X. Chen, "DCDB 2.0: a major update of the drug combination database," *Database (Oxford).*, vol. 2014, pp. 1–6, 2014.

[154] D. S. Wishart *et al.*, "HMDB 4.0: the human metabolome database for 2018," *Nucleic Acids Res.*, vol. 46, no. D1, pp. D608–D617, Jan. 2018.

[155] D. Wishart *et al.*, "T3DB: the toxic exposome database.," *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D928-34, Jan. 2015.

[156] T. Sterling and J. J. Irwin, "ZINC 15 - Ligand Discovery for Everyone," *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, 2015.

[157] R. D. Cramer, D. E. Patterson, and J. D. Bunce, "Comparative Molecular Field Analysis (CoMFA). 1. Effect of Shape on Binding of Steroids to Carrier Proteins," *J. Am. Chem. SOC*, vol. 110, pp. 5959–5967, 1988.

[158] J. Hert *et al.*, "Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures," *Org. Biomol. Chem.*, vol. 2, pp. 3256–3266, 2004.

[159] N. Huang, B. K. Shoichet, and J. J. Irwin, "Benchmarking Sets for Molecular Docking," *J. Med. Chem.*, vol. 49, no. 23, pp. 6789–6801, Nov. 2006.

[160] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet, "Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking," *J. Med. Chem.*, vol. 55, no. 14, pp. 6582–6594, Jul. 2012.

[161] “Tox21 Data Challenge 2014,” *https://tripod.nih.gov/tox21/challenge/, Accessed on March 20, 2018*. [Online]. Available: https://tripod.nih.gov/tox21/challenge/. [Accessed: 20-Mar-2018].

[162] Z. Wu *et al.*, “MoleculeNet: A benchmark for molecular machine learning,” *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, Mar. 2018.

[163] N. Lagarde, J.-F. Zagury, and M. Montes, “Benchmarking Data Sets for the Evaluation of Virtual Ligand Screening Methods: Review and Perspectives.,” *J. Chem. Inf. Model.*, vol. 55, no. 7, pp. 1297–307, Jul. 2015.

[164] J. Xia, E. L. Tilahun, T. E. Reid, L. Zhang, and X. S. Wang, “Benchmarking Methods and Data Sets for Ligand Enrichment Assessment in Virtual Screening,” *Methods*, vol. 71, pp. 146–157, 2015.

[165] T. U. Consortium, “UniProt: the universal protein knowledgebase,” *Nucleic Acids Res.*, vol. 45, no. November 2016, pp. 1–12, 2016.

[166] J. A. Blake *et al.*, “Gene ontology consortium: Going forward,” *Nucleic Acids Res.*, vol. 43, no. D1, pp. D1049–D1056, 2015.

[167] A. S. Rifaioglu *et al.*, “Large-scale automated function prediction of protein sequences and an experimental case study validation on PTEN transcript variants,” *Proteins Struct. Funct. Bioinforma.*, vol. 86, no. 2, pp. 135–151, 2017.

[168] T. Doğan *et al.*, “UniProt-DAAC: domain architecture alignment and classification, a new method for automatic functional annotation in UniProtKB,” *Bioinformatics*, vol. 32, no. 15, pp. 2264–2271, 2016.

[169] L. Lan, N. Djuric, Y. Guo, and S. Vucetic, “MS-kNN: protein function prediction by integrating multiple data sources.,” *BMC Bioinformatics*, vol. 14, no. S8, pp. 1–10, 2013.

[170] M. N. Wass, G. Barton, and M. J. E. Sternberg, “CombFunc: Predicting protein function using heterogeneous data sources,” *Nucleic Acids Res.*, vol.

40, no. W1, pp. 466–470, Jul. 2012.

[171] A. K. Tiwari and R. Srivastava, “A survey of computational intelligence techniques in protein function prediction.,” *Int. J. Proteomics*, vol. 2014, pp. 1–22, Jan. 2014.

[172] P. Koskinen, P. Törönen, J. Nokso-Koivisto, and L. Holm, “PANNZER: High-throughput functional annotation of uncharacterized proteins in an error-prone environment,” *Bioinformatics*, vol. 31, no. 10, pp. 1544–1552, 2015.

[173] Y. Jiang *et al.*, “An expanded evaluation of protein function prediction methods shows an improvement in accuracy,” *Genome Biol.*, vol. 17, no. 184, pp. 1–19, 2016.

[174] P. Radivojac *et al.*, “A large-scale evaluation of computational protein function prediction,” *Nat. Methods*, vol. 10, no. 3, pp. 221–229, 2013.

[175] J. A. Anderson, *An introduction to neural networks*. Boston, USA: MIT Press, 1995.

[176] G. Hinton *et al.*, “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[177] C. Angermueller, T. Pärnamaa, L. Parts, S. Oliver, O. Stegle, and S. Oliver, “Deep Learning for Computational Biology,” *Mol. Syst. Biol.*, vol. 12, no. 12, pp. 1–16, 2016.

[178] Y. Taigman, M. A. Ranzato, T. Aviv, and M. Park, “Deepface.” pp. 1–8, 2014.

[179] B. Ramsundar, P. Riley, D. Webster, D. Konerding, K. S. Edu, and P. S. Edu, “Massively Multitask Networks for Drug Discovery arXiv:1502.02072v1,” *arXiv*, pp. 1–27, 2015.

[180] Y. Bengio, *Learning Deep Architectures for AI*, vol. 2, no. 1. 2009.

[181] G. B. Goh, N. O. Hodas, and A. Vishnu, “Deep Learning for Computational

Chemistry," *arXiv*, vol. 1701.04503, pp. 1–50, 2017.

[182] X. L. Liu, "Deep Recurrent Neural Network for Protein Function Prediction from Sequence," *arXiv*, no. 1701.08318, pp. 1–38, 2017.

[183] R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, and Z. Chen, "ProLanGO: Protein function prediction using neural machine translation based on a recurrent neural network," *Molecules*, vol. 22, no. 10, 2017.

[184] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. October 2017, pp. 660–668, 2017.

[185] B. Szalkai, V. Grolmusz, and J. Hancock, "SECLAF: A Webserver and Deep Neural Network Design Tool for Hierarchical Biological Sequence Classification," *Bioinformatics*, vol. 34, pp. 2487–2489, 2018.

[186] A. Tavanaei *et al.*, "Towards Recognition of Protein Function based on its Structure using Deep Convolutional Networks," *IEEE Int. Conf. Bioinforma. Biomed.*, pp. 145–149, 2016.

[187] V. Gligorijević, M. Barot, and R. Bonneau, "deepNF: Deep network fusion for protein function prediction," *bioRxiv*, p. 223339, 2017.

[188] R. Fa, D. Cozzetto, C. Wan, and D. T. Jones, "Predicting Human Protein Function with Multi-task Deep Neural Networks," *bioRxiv*, vol. 256420, pp. 1–24, Jan. 2018.

[189] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," *Proc. 5th ACM Conf. Bioinformatics, Comput. Biol. Heal. Informatics - BCB '14*, pp. 533–540, 2014.

[190] G. Zou, Xianchun; Wang and Guoxian, "Protein Function Prediction Using Deep Restricted Boltzmann Machines," *BioMed Res. Int. Vol.*, vol. 2017, no. 1729301, pp. 1–9, 2017.

[191] A. S. Rifaioglu, T. Doğan, M. J. Martin, R. Cetin-Atalay, and V. Atalay, "DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks," *Sci. Rep.*, vol. 9, no. 7344, pp. 1–16, 2019.

[192] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets.," *Neural Comput.*, vol. 18, no. 7, pp. 1527–54, 2006.

[193] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[194] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[195] M. Windows *et al.*, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv*, vol. 1011.1669v, pp. 1–9, 2014.

[196] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv*, pp. 1–15, 2014.

[197] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *Proc. Mach. Learn. Res.*, vol. 28, no. 3, pp. 1139–1147, 2013.

[198] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *ArXiv*, vol. 1, no. 212, pp. 1–19, 2015.

[199] J. Shen *et al.*, "Predicting protein-protein interactions based only on sequences information.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 11, pp. 4337–41, Mar. 2007.

[200] K.-C. Chou, "Prediction of Protein Cellular Attributes Using Pseudo- Amino Acid Composition," *Proteins Struct., Funct., Genet.*, vol. 255, no. January, pp. 246–255, 2001.

[201] O. S. Sarac, O. Gürsoy-Yüzügüllü, R. Cetin-Atalay, and V. Atalay,

"Subsequence-based feature map for protein function classification.," *Comput. Biol. Chem.*, vol. 32, no. 2, pp. 122–30, Apr. 2008.

[202] Y.-C. Wang, Y. Wang, Z.-X. Yang, and N.-Y. Deng, "Support vector machine prediction of enzyme function with conjoint triad feature and hierarchical context," *BMC Syst. Biol.*, vol. 5, no. Suppl 1, p. S6, 2011.

[203] Y.-C. Wang, X.-B. Wang, Z.-X. Yang, and N.-Y. Deng, "Prediction of enzyme subfamily class via pseudo amino acid composition by incorporating the conjoint triad feature.," *Protein Pept. Lett.*, vol. 17, no. 11, pp. 1441–1449, 2010.

[204] Z.-H. You, K. C. C. Chan, and P. Hu, "Predicting Protein-Protein Interactions from Primary Protein Sequences Using a Novel Multi-Scale Local Feature Representation Scheme and the Random Forest," *PLoS One*, vol. 10, no. 5, p. e0125811, 2015.

[205] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences," *Nucleic Acids Res.*, vol. 36, no. 9, pp. 3025–3030, 2008.

[206] W. R. Qiu, X. Xiao, W. Z. Lin, and K. C. Chou, "IMethyl-PseAAC: Identification of protein methylation sites via a pseudo amino acid composition approach," *Biomed Res. Int.*, vol. 2014, 2014.

[207] Y. Xu, X. Wen, L. S. Wen, L. Y. Wu, N. Y. Deng, and K. C. Chou, "INitro-Tyr: Prediction of nitrotyrosine sites in proteins with general pseudo amino acid composition," *PLoS One*, vol. 9, no. 8, 2014.

[208] B. Liu, S. Wang, and X. Wang, "DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation," *Sci. Rep.*, vol. 5, no. September, pp. 1–11, 2015.

[209] I. Limongelli, S. Marini, and R. Bellazzi, "PaPI: pseudo amino acid composition to score human protein-coding variants," *BMC Bioinformatics*,

vol. 16, no. 1, p. 123, 2015.

[210] K. C. Chou, “Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes,” *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2005.

[211] N. Xiao, D. S. Cao, M. F. Zhu, and Q. S. Xu, “Protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences,” *Bioinformatics*, vol. 31, no. 11, pp. 1857–1859, 2015.

[212] “CAFA2 GitHub Repository.” [Online]. Available: https://github.com/yuxjiang/CAFA2. [Accessed: 14-Aug-2017].

[213] D. Cozzetto, F. Minneci, H. Currant, and D. T. Jones, “FFPred 3: Feature-based function prediction for all Gene Ontology domains,” *Sci. Rep.*, vol. 6, no. May, pp. 1–11, 2016.

[214] Q. Gong, W. Ning, and W. Tian, “GoFDR: A sequence alignment based method for predicting protein functions,” *Methods*, vol. 93, pp. 3–14, 2016.

[215] C. K. Stover *et al.*, “Complete genome sequence of Pseudomonas aeruginosa PAO1, an opportunistic pathogen,” *Nature*, vol. 406, no. 6799, pp. 959–964, 2000.

[216] J. W. Costerton, S. PS, and G. EP, “Bacterial Biofilms: A common cause of persistent infections,” *Science (80-.).*, vol. 284, pp. 1318–1323, 1999.

[217] R. M. Donlan and J. W. Costerton, “Biofilms: survivalmechanisms of clinically relevant microorganisms.,” *Clin.Microbiol. Rev.*, vol. 15, no. 2, pp. 167–19, 2002.

[218] D. A. Ryjenkov, M. Tarutina, O. V. Moskvin, and M. Gomelsky, “Cyclic diguanylate is a ubiquitous signaling molecule in bacteria: Insights into biochemistry of the GGDEF protein domain,” *J. Bacteriol.*, vol. 187, no. 5, pp. 1792–1798, 2005.

[219] A. Ueda and T. K. Wood, “Connecting quorum sensing, c-di-GMP, pel

polysaccharide, and biofilm formation in Pseudomonas aeruginosa through tyrosine phosphatase TpbA (PA3885)," *PLoS Pathog.*, vol. 5, no. 6, pp. 1–15, 2009.

[220] C.-Y. Chang, "Surface Sensing for Biofilm Formation in Pseudomonas aeruginosa," *Front. Microbiol.*, vol. 8, no. January, pp. 1–8, 2018.

[221] R. P. Ryan *et al.*, "HD-GYP domain proteins regulate biofilm formation and virulence in Pseudomonas aeruginosa," *Environ. Microbiol.*, vol. 11, no. 5, pp. 1126–1136, 2009.

[222] N. Zhou *et al.*, "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome Biol.*, vol. 20, no. 244, pp. 1–23, 2019.

[223] G. J. P. Van Westen *et al.*, "Benchmarking of protein descriptor sets in proteochemometric modeling (part 1): comparative study of 13 amino acid descriptor sets," *J. Cheminform.*, vol. 5, no. 41, pp. 1–11, 2013.

[224] C. Szegedy *et al.*, "Going deeper with convolutions," *arXiv*, vol. arXiv:1409, pp. 1–12, 2014.

[225] D. Mendez *et al.*, "ChEMBL: towards direct deposition of bioassay data," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D930–D940, 2018.

[226] Y. Wang *et al.*, "PubChem BioAssay: 2017 update," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D955–D963, 2017.

[227] J. L. Reymond, "The Chemical Space Project," *Acc. Chem. Res.*, vol. 48, no. 3, pp. 722–730, 2015.

[228] G. Cano *et al.*, "Automatic selection of molecular descriptors using random forest: Application to drug discovery," *Expert Syst. Appl.*, vol. 72, pp. 151–159, 2017.

[229] H. Yu *et al.*, "A Systematic Prediction of Multiple Drug-Target Interactions

from Chemical , Genomic , and Pharmacological Data," *PLoS One*, vol. 7, no. 5, pp. 1–14, 2012.

[230] D. Emig *et al.*, "Drug Target Prediction and Repositioning Using an Integrated Network-Based Approach," *PLoS One*, vol. 8, no. 4, pp. 1–17, 2013.

[231] E. B. Lenselink *et al.*, "Beyond the hype: Deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set," *J. Cheminform.*, vol. 9, no. 45, pp. 1–14, 2017.

[232] A. Gonczarek, J. M. Tomczak, S. Zaręba, J. Kaczmar, P. Dąbrowski, and M. J. Walczak, "Interaction prediction in structure-based virtual screening using deep learning," *Comput. Biol. Med.*, vol. 100, no. September, pp. 253–258, 2017.

[233] E. N. Feinberg *et al.*, "PotentialNet for Molecular Property Prediction.," *ACS Cent. Sci.*, vol. 4, no. 11, pp. 1520–1530, 2018.

[234] Z. Wu *et al.*, "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018.

[235] B. Liu *et al.*, "Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models," *ACS Cent. Sci.*, vol. 3, no. 10, pp. 1103–1113, 2017.

[236] T. B. Hughes, N. Le Dang, G. P. Miller, and S. J. Swamidass, "Modeling reactivity to biological macromolecules with a deep multitask network," *ACS Cent. Sci.*, vol. 2, no. 8, pp. 529–537, 2016.

[237] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, "Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks," *ACS Cent. Sci.*, vol. 4, no. 1, pp. 120–131, Jan. 2018.

[238] R. Gómez-Bombarelli *et al.*, "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules," *ACS Cent. Sci.*, vol. 4, no. 2, pp. 268–276, 2018.

[239] S. Pushpakom *et al.*, "Drug repurposing: progress, challenges and recommendations," *Nat. Rev. Drug Discov.*, vol. 18, pp. 41–58, 2019.

[240] M. Ragoza, L. Turner, and D. R. Koes, "Ligand Pose Optimization with Atomic Grid-Based Convolutional Neural Networks," *arXiv*, vol. 1710.07400, pp. 1–10, 2017.

[241] M. Ragoza, 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.

[242] J. Hochuli, A. Helbling, T. Skaist, M. Ragoza, and D. R. Koes, "Visualizing convolutional neural network protein-ligand scoring," *J. Mol. Graph. Model.*, vol. 84, pp. 96–108, 2018.

[243] J. Sunseri, J. E. King, P. G. Francoeur, and D. R. Koes, "Convolutional neural network scoring and minimization in the D3R 2017 community challenge," *J. Comput. Aided. Mol. Des.*, vol. 33, no. 1, pp. 19–34, 2019.

[244] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a Next-Generation Open Source Framework for Deep Learning," *arXiv*, vol. 1908.00213, pp. 1–6, 2019.

[245] B. Ramsundar, P. Eastman, P. Walters, and V. Pande, *Deep Learning for the Life Sciences : Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. .

[246] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry," *arXiv*, vol. 1905.13741, pp. 1–16, 2019.

[247] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, 2010.

[248] D. Duvenaud, D. Maclaurin, J. Aguilera-iparraguirre, G. Rafael, T. Hirzel, and R. P. Adams, "Convolutional Networks on Graphs for Learning

Molecular Fingerprints," *arXiv*, vol. arXiv:1509, pp. 1–9, 2015.

[249] M. Fernandez *et al.*, "Toxic Colors: The Use of Deep Learning for Predicting Toxicity of Compounds Merely from Their Graphic Images," *J. Chem. Inf. Model.*, vol. 58, pp. 1533–1543, 2018.

[250] G. B. Goh, C. Siegel, N. Hodas, and A. Vishnu, "Using Rule-Based Labels for Weak Supervised Learning A ChemNet for Transferable Chemical Property Prediction," in *KDD '18 Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, vol. 1, pp. 302–310.

[251] G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas, and N. Baker, "Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models," *arXiv*, vol. arXiv:1706, pp. 1–38, 2017.

[252] G. Landrum, "Rdkit: Open-source cheminformatics software," *URL http://www. rdkit. org/, https://github. com/rdkit/rdkit*, 2016. [Online]. Available: http://www.rdkit.org/. [Accessed: 12-Nov-2018].

[253] "RDKit Generating Depictions." [Online]. Available: https://www.rdkit.org/docs/GettingStartedInPython.html#working-with-2d-molecules-generating-depictions. [Accessed: 04-Mar-2019].

[254] N. Bosc, F. Atkinson, E. Felix, A. Gaulton, A. Hersey, and A. R. Leach, "Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery," *J. Cheminform.*, vol. 11, no. 4, pp. 1–16, 2019.

[255] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "{G}radient-based {L}earning {A}pplied to {D}ocument {R}ecognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[256] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "ImageNet Classification

with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst. 25*, vol. 1, pp. 1097–1105, 2012.

[257] "TFLearn: Deep learning library featuring a higher-level API for TensorFlow," 2018. [Online]. Available: https://github.com/tflearn/tflearn%7D.

[258] G. W. Bemis and M. A. Murcko, "The properties of known drugs. 1. Molecular frameworks," *J. Med. Chem.*, vol. 39, no. 15, pp. 2887–2893, 1996.

[259] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet, "Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking," *J. Med. Chem.*, vol. 55, no. 14, pp. 6582–6594, Jul. 2012.

[260] S. G. Rohrer and K. Baumann, "Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data," *J. Chem. Inf. Model.*, vol. 49, no. 2, pp. 169–184, 2009.

[261] R. Meine *et al.*, "Indole-3-carbonitriles as DYRK1A inhibitors by fragment-based drug design," *Molecules*, vol. 23, no. 2, pp. 1–23, 2018.

[262] M. Drexel, J. Kirchmair, and S. Santos-Sierra, "INH14, a small-molecule urea derivative, inhibits the IKKα/β-dependent TLR inflammatory response," *ChemBioChem*, pp. 1–17, 2018.

[263] A. L. Mitchell *et al.*, "InterPro in 2019 : improving coverage , classification and access to protein sequence annotations," *Nucleic Acids Res.*, vol. 47, no. Database Issue, pp. 351–360, 2019.

[264] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. Mckusick, "Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders," *Nucleic Acids Res.*, vol. 33, no. Database, pp. 514–517, 2005.

[265] A. Rath *et al.*, "Representation of rare diseases in health information systems: The orphanet approach to serve a wide range of end users," *Hum. Mutat.*, vol.

33, no. 5, pp. 803–808, 2012.

[266] A. S. Rifaioglu, E. Nalbat, V. Atalay, M. J. Martin, R. Cetin-Atalay, and T. Doğan, "DEEPScreen: high performance drug–target interaction prediction with convolutional neural networks using 2-D structural compound representations," *Chem. Sci.*, vol. 11, no. 9, pp. 2531–2557, 2020.

[267] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," *2018 Int. Interdiscip. PhD Work.*, pp. 117–122, 2018.

[268] N. Thomas *et al.*, "Tensor field networks : Rotation- and translation-equivariant neural networks for 3D point clouds," *arXiv*, vol. 1802.08219, pp. 1–19, 2018.

[269] L. Chen *et al.*, "Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening," *PLoS One*, vol. 14, no. 8, p. e0220113, 2019.

[270] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005, vol. I, pp. 886–893.

[271] S. Park, Y. Kwon, H. Jung, S. Jang, H. Lee, and W. Kim, "CSgator : an integrated web platform for compound set analysis," *J. Cheminform.*, pp. 4–11, 2019.

[272] J. Hastings *et al.*, "ChEBI in 2016 : Improved services and an expanding collection of metabolites," *Nucleic Acids Res.*, vol. 44, no. Datanase issue, pp. 1214–1219, 2016.

[273] "The Anatomical Therapeutic Chemical (ATC) Classification System." [Online]. Available: http://www.whocc.no/atc/structure_and_principles/. [Accessed: 12-Nov-2016].

[274] T. Gene and O. Consortium, “The Gene Ontology Resource : 20 years and still GOing strong,” *Nucleic Acids Res.*, vol. 47, no. November 2018, pp. 330–338, 2019.

[275] P. Erwin and W. Perkins, *Medline: A Guide to Effective Searching in PubMed & Other Interfaces*, vol. 2nd Editio, no. 107. 2007.

[276] L. M. Schriml *et al.*, “Disease Ontology : a backbone for disease semantic integration,” *Nucleic Acids Res.*, vol. 40, no. Database issue, pp. 940–946, 2012.

[277] P. Moreno *et al.*, “BiNChE : A web tool and library for chemical enrichment analysis based on the ChEBI ontology,” *BMC Bioinformatics*, vol. 16, no. 56, pp. 1–7, 2015.

[278] C. Huang *et al.*, “The DrugPattern tool for drug set enrichment analysis and its prediction for bene fi cial effects of oxLDL on type 2 diabetes,” *J. Genet. Genomics*, vol. 45, no. 7, pp. 389–397, 2018.

[279] H. Ding, H. Mamitsuka, and S. Zhu, “Similarity-based machine learning methods for predicting drug ^ target interactions : a brief review,” *Brief. Bioinform.*, vol. 15, no. 5, pp. 734–747, 2013.

[280] H. Lim, P. Gray, L. Xie, and A. Poleksic, “Improved genome-scale multi-target virtual screening via a novel collaborative filtering approach to cold-start problem,” *Sci. Rep.*, vol. 6, no. 38860, pp. 1–11, Dec. 2016.

[281] R. S. Olayan, H. Ashoor, and V. B. Bajic, “DDR: Efficient computational method to predict drug-Target interactions using graph mining and machine learning approaches,” *Bioinformatics*, vol. 34, no. 7, pp. 1164–1173, 2018.

[282] M. Lee, H. Kim, H. Joe, and H.-G. Kim, “Multi-channel PINN: investigating scalable and transferable neural networks for drug discovery,” *J. Cheminform.*, vol. 11, no. 46, pp. 1–16, 2019.

[283] I. Lee, J. Keum, and H. Nam, “DeepConv-DTI: Prediction of drug-target

interactions via deep learning with convolution on protein sequences," *PLoS Comput. Biol.*, vol. 15, no. 6, pp. 1–21, 2019.

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

[285] M. Karimi, D. Wu, and Z. Wang, "DeepAffinity : Interpretable Deep Learning of Compound-Protein Affinity through Unified Recurrent and Convolutional Neural Networks," *Bioinformatics*, pp. 1–8, 2019.

[286] M. Kukiełka, M. M. Stepniewska-dziubinska, and P. Siedlecki, "Development of a protein – ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions," *Bioinformatics*, vol. 35, no. 8, pp. 1334–1341, 2019.

[287] M. M. Stepniewska-dziubinska, P. Zielenkiewicz, and P. Siedlecki, "Development and evaluation of a deep learning model for protein – ligand binding affinity prediction," *Bioinformatics*, vol. 34, no. 21, pp. 3666–3674, 2018.

[288] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug Discov. Today*, vol. 23, no. 6, pp. 1241–1250, 2018.

[289] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges.," *Brief. Bioinform.*, May 2017.

[290] M. Thafar, A. Bin Raies, S. Albaradei, M. Essack, and V. B. Bajic, "Comparison Study of Computational Prediction Tools for Drug-Target Binding Affinities," *Front. Chem.*, vol. 7, no. November, p. 782, 2019.

[291] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester, "SimBoost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines," *J. Cheminform.*, vol. 9, no. 1, pp. 1–14, 2017.

[292] B. Steiner *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *NeuroIPS*, 2019, no. NeurIPS, pp. 1–12.

[293] S. K. Burley, H. M. Berman, G. J. Kleywegt, J. L. Markley, H. Nakamura, and S. Velankar, “Protein Data Bank (PDB): The single global macromolecular structure archive,” *Methods Mol. Biol.*, vol. 1607, pp. 627–641, 2017.

[294] Z. Liu *et al.*, “PDB-wide collection of binding data: Current status of the PDBbind database,” *Bioinformatics*, vol. 31, no. 3, pp. 405–412, 2015.

[295] C. Zhang and S. H. Kim, “Environment-dependent residue contact energies for proteins,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 97, no. 6, pp. 2550–2555, 2000.

[296] S. Henikoff and J. G. Henikoff, “Amino acid substitution matrices from protein blocks.,” *Proc. Natl. Acad. Sci.*, vol. 89, no. 22, pp. 10915–10919, 1992.

[297] R. Grantham, “Amino acid difference formula to help explain protein evolution,” *Science (80-.).*, vol. 185, no. 4154, pp. 862–864, 1974.

[298] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff, and D. Baker, “Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins,” *Proteins Struct. Funct. Genet.*, vol. 34, no. 1, pp. 82–95, 1999.

[299] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *ICML’10 Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, pp. 807–814, 2010.

[300] A. Airola and T. Pahikkala, “Fast Kronecker Product Kernel Methods via Generalized Vec Trick,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3374–3387, 2018.

[301] M. Gönen and G. Heller, “Concordance probability and discriminatory power

in proportional hazards regression," *Biometrika*, vol. 92, no. 4, pp. 965–970, 2005.

[302] G. Rodgers *et al.*, "Glimmers in illuminating the druggable genome," *Nat. Rev. Drug Discov.*, vol. 17, no. 5, pp. 301–302, 2018.

[303] V. Joshi *et al.*, "CROssBAR: Comprehensive Resource of Biomedical Relations with Network Representations and Deep Learning," in *Bio-Ontologies COSI at ISMB/ECCB 2019*, 2019.

[304] M. Yang *et al.*, "Linking drug target and pathway activation for effective therapy using multi-Task learning," *Sci. Rep.*, vol. 8, no. 1, pp. 1–10, 2018.

[305] M. Gütlein, A. Karwath, and S. Kramer, "CheS-Mapper 2.0 for visual validation of (Q)SAR models," *J. Cheminform.*, vol. 6, no. 41, pp. 1–18, 2014.

[306] D. S. Karlov, S. Sosnin, I. V. Tetko, and M. V. Fedorov, "Chemical space exploration guided by deep neural networks," *RSC Adv.*, no. 9, pp. 5151–5157, 2019.

[307] A. P. A. Janssen *et al.*, "Drug Discovery Maps, a Machine Learning Model That Visualizes and Predicts Kinome-Inhibitor Interaction Landscapes," *J. Chem. Inf. Model.*, vol. 59, no. 3, pp. 1221–1229, 2019.

[308] M. Awale and J. L. Reymond, "Web-based 3D-visualization of the DrugBank chemical space," *J. Cheminform.*, vol. 8, no. 25, pp. 1–8, 2016.

[309] H. A. Gaspar, I. I. Baskin, G. Marcou, D. Horvath, and A. Varnek, "Chemical Data Visualization and Analysis with Incremental Generative Topographic Mapping: Big Data Challenge," *J. Chem. Inf. Model.*, vol. 55, no. 1, pp. 84–94, 2015.

[310] G. van der Maaten, Laurens; Hinton, "Visualizing Data using t-SNE Laurens," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605.

[311] Bokeh Development Team, "Bokeh: Python library for interactive

visualization," 2014. [Online]. Available: http://www.bokeh.pydata.org. [Accessed: 03-Mar-2019].

[312] L. H. Mervin, A. M. Afzal, G. Drakakis, R. Lewis, O. Engkvist, and A. Bender, "Target prediction utilising negative bioactivity data covering large chemical space," *J. Cheminform.*, vol. 7, no. 1, p. 51, Dec. 2015.

[313] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, 2018.

[314] R. C. Fong and A. Vedaldi, "Interpretable Explanations of Black Boxes by Meaningful Perturbation..," in *International Conference on Computer Vision*, 2017, pp. 3429–3437.

[315] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020.

[316] J. Jiménez-Luna, F. Grisoni, and G. Schneider, "Drug discovery with explainable artificial intelligence," *arXiv*, vol. 2007.00523, pp. 1–15, 2020.

[317] K. Yingkai Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, and P. Zhang, "Interpretable drug target prediction using deep neural representation," in *IJCAI International Joint Conference on Artificial Intelligence*, 2018, vol. 2018-July, pp. 3371–3377.

[318] K. Preuer, G. Klambauer, F. Rippmann, S. Hochreiter, and T. Unterthiner, "Interpretable Deep Learning in Drug Discovery," *arXiv*, vol. 1903, pp. 1–17, 2019.

[319] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image," *J. Mach. Learn. Res.*, vol. 37, pp. 1–8, 2015.

[320] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, pp. 4078–4088,

2017.

[321] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” *Adv. Neural Inf. Process. Syst.*, pp. 3637–3645, 2016.

[322] L. Chen, J. Lu, N. Zhang, T. Huang, and Y.-D. Cai, “A hybrid method for prediction and repositioning of drug Anatomical Therapeutic Chemical classes.,” *Mol. Biosyst.*, vol. 10, no. 4, pp. 868–77, 2014.

[323] F. S. Chen and Z. R. Jiang, “Prediction of drug’s Anatomical Therapeutic Chemical (ATC) code by integrating drug-domain network,” *J. Biomed. Inform.*, vol. 58, pp. 80–88, 2015.

[324] S. K. Mohamed, A. Nounu, and V. Nováček, “Drug target discovery using knowledge graph embeddings,” *Proc. ACM Symp. Appl. Comput.*, vol. Part F1477, pp. 11–18, 2019.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Rifaioğlu, Ahmet Süreyya
Nationality: Turkish (TC)
Date and Place of Birth: 27 May 1988, Libya
Marital Status: Married
Phone: +90 312 210 55 32
email: arifaioglu@ceng.metu.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Computer Engineering	2015
BS	Doğuş University Computer Engineering	2010
High School	Osman Ötken Anatolian High School, Hatay	2006

WORK EXPERIENCE

Year	Place	Enrollment
2011-Present	METU Dept. of Computer Eng.	Teaching/Research Assistant
2019 Apr. - 2020 Feb.	EMBL – European Bioinformatics Institute	Predoctoral Visiting Researcher
2017 Nov. – 2018 Feb.	EMBL – European Bioinformatics Institute	Predoctoral Visiting Researcher
2014 Oct. – 2015 Jan.	EMBL – European Bioinformatics Institute	Predoctoral Visiting Researcher
2011 June-2011 Aug.	ISIS Information Technologies	Software Engineer – Jr. SAP Consultant

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1. **Rifaioglu, A.S.**, Cetin-Atalay, R., Kahraman, D.C., Doğan, T., Martin, M. J., and Atalay, V., "MDeePred: Multi-Channel Deep Chemogeomic Prediction of Binding Affinity in Drug Discovery," Under Review in a peer-reviewed journal

2. Donmez, Ataberk, **Rifaioglu, A.S.**, Acar, A. C., Doğan T., Cetin-Atalay R. and Atalay M., "iBioProVis: Interactive Visualization and Analysis of Compound Bioactivity," Bioinformatics, btaa496, (2020). https://doi.org/10.1093/bioinformatics/btaa496

3. **Rifaioglu, A. S.,** Nalbat, E., Atalay, V., Martin, M. J., Cetin-Atalay, R. and Doğan, T. "DEEPScreen: high performance drug–target interaction prediction with convolutional neural networks using 2-D structural compound representations," Chemical Science, Ml, 1–27, (2020). http://doi.org/10.1039/c9sc03414e

4. Zhou, N., Jiang, Y., Bergquist, T.R., Lee, A.J., Kacsoh, B.Z., Crocker, A.W., Lewis, K.A., Georghiou, G., Nguyen, H.N., Hamid, M.N., Davis, L., Doğan, T., Atalay, V., **Rifaioglu A.S.**, et al., "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," Genome Biology, 20(244), 1–23, (2019). http://doi.org/10.1186/s13059-019-1835-8

5. **Rifaioglu, A.S.**, Doğan, T., Martin, M. J., Cetin-Atalay, R., and Atalay, V. DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks, Scientific Reports, Nature, 9(7344), 1-16, (2019), https://doi.org/10.1038/s41598-019-43708-3

6. Dalkiran, A., **Rifaioglu, A. S.**, Martin, M. J., Cetin-Atalay, R., Atalay, V. and Doğan, T., ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature, BMC Bioinformatics, 19(334), 1-13, (2018), https://doi.org/10.1186/s12859-018-2368-y

7. **Rifaioglu, A.S.**, Atas, H., R., Martin, M. J., Cetin- Atalay, R., Atalay, M. and Doğan, T. Recent Applications of Machine Intelligence including Deep Learning on Virtual Screening: Methods, Tools and Databases. Briefings in Bioinformatics, 20(5), 1878-1912, (2019), https://doi.org/10.1093/bib/bby061

8. **Rifaioglu, A.S.**, Doğan, T., Saraç, Ö. S., Ersahin, T., Saidi, R., Atalay, M. V., Martin, M. J. and Cetin- Atalay, R.. Large- scale automated function prediction of protein sequences and an experimental case study validation on PTEN transcript variants. Proteins: Structure, Function, and Bioinformatics, 86(2), 135-151, (2018), https://doi.org/10.1002/prot.25416

TALKS and POSTERS in PEER-REVIEWED CONFERENCE

1. Joshi, V., **Rifaioglu, A.S.,** Dogan, T, Ataş, H., Sinoplu, E., Nighingale, A., Volynkin, V., Zellner, H., Saidi, R., Cetin-Atalay, R., Martin, M.J. & Atalay, M.V. (2019). CROssBAR: Comprehensive Resource of Biomedical Relations with Network Representations and Deep Learning. *ISMB/ECCB 2019: 27th Annual International Conference on Intelligent Systems for Molecular Biology*, July 21-25, 2019, Basel, Switzerland. (**Oral Presentation**)

2. **Rifaioglu, A.S.**, Dönmez, A., Acar, A.C., Martin, M.J., Dogan, T., Cetin-Atalay, R., Atalay, V**,** iBioProVis: Interactive Visualization and Analysis of Compound Bioactivity Space**,** *Biological Data Visualization (BioVis:) COSI at ISMB/ECCB 2019*. (**Oral Presentation**)

3. **Rifaioglu, A.S.**, Atalay, M.V., Martin, M.J., Cetin-Atalay, R. & Doğan, T. (2018). DEEPScreen: Drug-Target Interaction Prediction with Deep Convolutional Neural Networks Using Compound Images. *ISMB 2018: 26th Annual International Conference on Intelligent Systems for Molecular Biology*, July 6-10, 2018, Chicago, USA. (**Oral Presentation**)

4. Doğan, T., **Rifaioglu, A.S.**, Saidi, R., Martin, M., Atalay, M.V. & Cetin-Atalay, R. (2018). Automated Negative Gene Ontology Based Functional Predictions for Proteins with UniGOPred. *ISMB 2018: 26th Annual International Conference on Intelligent Systems for Molecular Biology*, July 6-10, 2018, Chicago, USA. (**Oral Presentation**)

5. **Rifaioglu, A.S.**, Martin, M.J., Cetin-Atalay, R., Atalay, M.V. & Doğan, T., Investigation of Multi-task Deep Neural Networks for Protein Function Prediction, *ISMB/ECCB 2017: 25th Annual International Conference on Intelligent Systems for Molecular Biology, Function- COSI***,** July 21 - July 25, 2017, Prague, Czech Republic doi: 10.7490/f1000research.1114653.1 (**Oral Presentation**)

6. **Rifaioglu, A.S.**, Doğan, T., Sarac, Ö.S., Saidi, R., Atalay, V., Martin, M. J. & Atalay, R. (2017). UniGOPred: A Large-Scale Automated GO Term Annotation System for UniProtKB. *GLBIO 2017: Great Lakes Bioinformatics Conference*, May 15-17, 2017, Chicago, USA. (**Poster – Best Poster Award**)

7. **Rifaioglu, A.S.**, Doğan, T., Saraç, Ö. S., Atalay, V., Martin, M. & Atalay, R. (2015). UniGOPred and ECPred: Automated Function Prediction Tools Based on A Combination of Different Classifiers. *AFP-CAFA SIG, ISMB/ECCB 2015: 23rd Annual International Conference on Intelligent Systems for Molecular Biology,* July 10-14, 2015, Dublin, Republic of Ireland. (**Poster**)

8. Doğan, T., Ersahin, T., **Rifaioglu, A.S.**, Poggioli, D., Nightingale, A., Martin, M. & Cetin-Atalay, R. (2015). Computational drug target prediction and validation in PI3K/AKT pathway. *ISMB/ECCB 2015: 23th Annual International Conference on Intelligent Systems for Molecular Biology,* July 2015, Dublin, Republic of Ireland. (**Poster**)

9. **Rifaioglu, A.S.**, Doğan, T. & Can, T. (2015). Unsupervised identification of redundant domain entries in InterPro database using clustering techniques. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics* (pp. 505-506), ACM, September 9-12, 2015, Atlanta, USA. (**Poster**)