Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Bonggun Shin Date

Deep Learning Approaches Towards Computerized Drug Discovery

By

Bonggun Shin
Doctor of Philosophy

Computer Science and Informatics

Joyce C. Ho, Ph.D.
Advisor

Li Xiong, Ph.D.
Committee Member

Zhaohui S. Qin, Ph.D.
Committee Member

Jimeng Sun, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Deep Learning Approaches Towards Computerized Drug Discovery

By

Bonggun Shin
B.A., Illinois Institute of Technology, IL, 2006
M.Sc., Korea Advanced Institute of Science and Technology, South Korea, 2010
M.Sc., Emory University, GA, 2019

Advisor: Joyce C. Ho, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2020

Abstract

Deep Learning Approaches Towards Computerized Drug Discovery
By Bonggun Shin

Proposing a new drug candidate is an essential part of the drug discovery process, consisting of many sub-tasks. Traditionally, these tasks have been tackled by chemistry and pharmaceutical experts and take years to design. Therefore, this thesis aims to accelerate drug discovery by proposing deep-learning models that accomplishes these tasks effectively and quickly. For the target identification problem, we propose new feature selection methods for both disease-related and prognosis-related features. Next, we propose a new drug-target interaction model to perform the drug re-purposing task. In this model, we present a new molecule representation to overcome the limitation of the current models. We also propose a novel drug candidate generation model that can modify an existing drug to meet given molecule properties. For each project, we present an empirical evaluation to show the competency of the proposed approaches. In addition, we also provide analyses or case studies to demonstrate the practicality of our approaches.

Deep Learning Approaches Towards Computerized Drug Discovery

By

Bonggun Shin
B.A., Illinois Institute of Technology, IL, 2006
M.Sc., Korea Advanced Institute of Science and Technology, South Korea, 2010
M.Sc., Emory University, GA, 2019

Advisor: Joyce C. Ho, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2020

Acknowledgments

He knoweth the way that I take: when He hath tried me, I shall come forth as gold.

– Book of Job 23:10

This work represents the realization of my lifelong goal, and the way to it has never been smooth. Thankfully, I have been guided and supported by many remarkable people, and without them, this work could not be realized as it is now. Although my appreciation that I hold for them cannot be fully expressed in just a short text, I'd like to remember them throughout my life by frequently reminding these brief acknowledgments.

I am profoundly grateful to my advisor Prof. Joyce Ho for her mentorship, encouragement, and supports. She encouraged me to explore new research areas, gave me insights, and waited for me so that I can overcome hardships by myself. I am extremely blessed to have been a graduate student of such a wonderful academic mentor.

I would like to extend my thanks to the members of my dissertation committee: Prof. Lee Xiong, Prof. Steve Qin, and Prof. Jimeng Sun. for helping me to improve the quality of my dissertation through valuable comments and feedbacks.

Besides the committee members, I would like to also thank Prof. Jinho Choi. For the earlier years of Emory, he trained me to be a better researcher. Thanks to his training, my programming and academic writing have been dramatically improved.

I also want to express my gratitude to the excellent collaborators, all members of Deargen. They have introduced me to interesting real-world problems and supported me to solve them together. The majority parts of this dissertation would not be possible without this collaboration.

I would like to give a special thanks to my soul mate, Kim Heonoo and Sunho

Hwang. I'll remember your prayers, supports, and love especially in the midst of hardships.

Finally, this dissertation is dedicated to my parents, Seunghae Shin and Yongsoon Kim; my sister, Eunkyung Shin, my brothers, Seokki Kwon and Keumku Kang; my son, Caleb Taeha Shin, and lastly but most importantly my wife, Junghyun Kang. Your unwavering love has been the motivation of my life and this work. I'm the most fortunate one in the world because I met all of you. Thank you and I love you all.

– *Bonggun Shin*

Emory University

March 2020

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

> *If even one new drug of the stature of penicillin or digitalis has been unjustifiably banished to a company's back shelf because of exceedingly stringent regulatory requirements, that event will have harmed more people than all the toxicity that has occurred in the history of modern drug development.*
>
> – *William Wardell*

Drug discovery is a challenging, time-consuming and costly process, which takes an average of 9 to 12 years to develop a single drug [42]. Any failure in this costly and lengthy process can lead to enormous financial losses. What's worse is failures occur quite often throughout the drug development pipeline. According to [43], the estimated average cost to develop a new medicine and gain FDA approval is $1.4 billion. Among this amount, 40% of it is spent on the candidate compound genera-

	Candidates	Pre-clinical	Phase 1	Phase 2	Phase 3
Time	4-5 years	1-2 years	1-2 years	1-2 years	2-3 years
Cost	$550M	$125M	$225M	$250M	$250M
Molecules	5k-10k	10-20	5-10	2-5	1-2

Table 1.1: Traditional Drug Discovery and Development Process*. It consists of drug candidate generation step (Candidates), cell-line and animal experiment step (Pre-clinical), and three phase of human clinical tests (Phase 1,2, and 3).

* The table is adapted from the slide of Data Mining for Drug Discovery in KDD19.

tion step as summarized in Table 1.1. In this step, around 5,000 to 10,000 molecules are generated as candidates but over 99.9% of them will be eventually discarded and only 0.1% of them will be approved to the market. This low approval rate is also attributed to the stringent regulations to protect people from unknown adversarial effects. However, as Wardell pointed out in the quote, it's important to continue to develop new drugs despite these difficulties. One of the efforts to overcome these challenges is adopting computer-aided systems in many subtasks of the drug discovery pipeline. With the help of intelligent and automated systems, subtasks of the drug discovery process can be optimized to make the entire process cost-effective. We first overview those subtasks in the pipeline (Section 1.1) and introduce the current trends of computer-aided drug discovery (Section 1.2). Inspired by the areas that have overcome the limitations and made many breakthroughs by applying deep learning, we discuss what innovations we have made in drug discovery through this work (Section 1.3).

1.1 Drug Discovery Pipeline

The drug discovery process involves several disciplines such as genomics, chemistry, biology, and pharmacology, which can be summarized as a series of five stages (Figure 1.1).

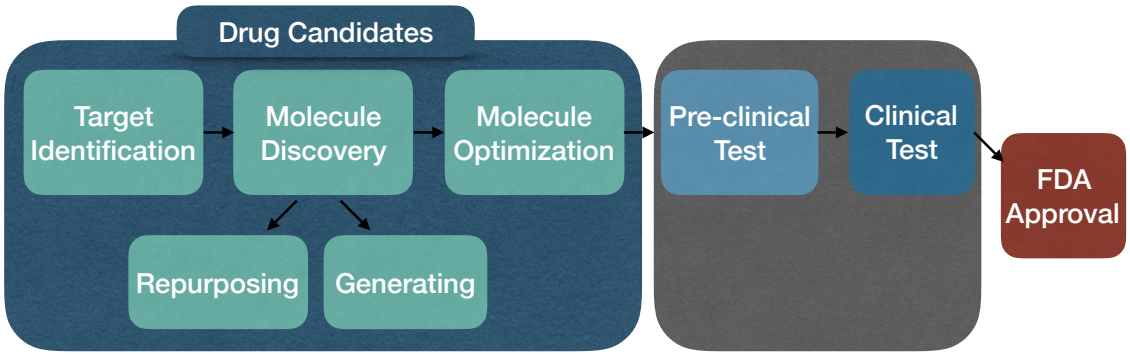


Figure 1.1: Drug discovery pipeline and the sub-tasks in the Drug Cadidates phase.

- **Target Identification**: The goal of this stage is to find biological entities (proteins, genes or RNAs) associated with a specific disease with which a putative drug interacts [4]. Found targets need to be validated using its ability to regulate biological processes. Through numerous studies, we confirm the relationships between the target and the disease state [123].

- **Molecule Discovery**: Molecule discovery involves the finding of synthetic chemicals that shows a degree of efficacy for a given target and potentially aims to cure diseases associated with the target [82]. Molecules can be divided into two types based on its molecular weights: small and large molecules. In this work, we refer to small molecules as molecules because they account for about 70% of all drugs[1]. When finding a molecule candidate, we can generate a new molecule (Section 5) or repurpose a known molecule to a new target (Section 4).

- **Molecule Optimization**: At the molecule optimization stage, we improve the potency and other important properties of a candidate molecule. The goal of the optimization is to prioritize and select promising candidates with safety and potency.

- **Pre-clinical Test**: These selected candidates are tested on animals for potency and toxicity before tested on human volunteers. Using in vivo animal studies, we also characterize toxicity profiles for various doses [147, 19].

- **Clinical Test**: The clinical test consists of three phases that investigate side-effects, safety, dosage profiles, potency and other properties of the candidate molecule on human volunteers [147]. If any drug passed all three stages of the clinical test, then it is approved by FDA and available to the market.

The advancement of computing technology has led to the accumulation of large biological entity databases with its bioactivity profiles [117, 83]. As such, the con-

[1] `https://www.dcatvci.org/5852-small-molecules-still-leading-in-new-drug-approvals`

ventional drug discovery methodology, which relies on manual works of expert groups is reaching their limit, because the dataset is now too large for humans to extract valuable information from it. This inefficient nature of the candidate generation step serves as motivation to consider an alternative method, artificial intelligence (AI) based drug discovery. The advantage of AI platforms is higher expected success rate by reducing the processing time because it can systematically generate candidate molecules and analyze immense amounts of chemical and biological data in a short period of time.

1.2 AI in Drug Discovery

Recently, pharmaceutical giants have started to collaborate with small companies actively developing drug discovery software using AI. As examples, Takeda Pharmaceutical is using Numerates AI platform to find small-molecules for oncology, gastroenterology, and central nervous system disorders. Pfizer integrates IBM Watson for Drug Discovery into its pipeline to effectively utilize Pfizer's scientific knowledge with a set of machine learning methods expecting the improved search for immuno-oncology drugs. Eli Lilly is supporting Atomwise to generate promising drug candidates using its deep learning-based molecule screening tool. One might be skeptical of this new approach because there have been few success stories, however, if shown to be effective, then this would become a new standard pipeline because it could save much time and cost. Not only that, just like many AIs have created innovation based on deep learning, new drug development methodologies can make a big breakthrough with deep learning.

In an image recognition task, for example, the performance of computer models was significantly behind the one of a human until 2010, when the ImageNet [39] project begun. In the ImageNet project, many people gathered, cleaned and annotated about

1.4 billion of images, with which many researchers have proposed advanced deep learning models. With the help of the well-curated dataset and appropriate computational power, many deep learning models can surpass human performance [65, 73, 129, 155]. Similar to computer vision, natural language process (NLP) has gone through a long dark period, however, since Google AI's BERT [41] and Transformer [160] have been proposed, deep learning models have achieved another level. There are two main factors to this success; availability of high-quality large data sets and the rapid development of powerful computational hardware.

These success stories have motivated drug discovery researchers to adopt advanced deep learning methods to one of the sub-tasks in the drug discovery pipeline. As described in Section 1.1, there are many sub-tasks in the pipeline, many of which have not adopted the recently developed deep learning techniques or still relied on manual labors of experts. For example, the previous state of the art of the drug toxicity prediction problem [181] had been based on an ensemble of three traditional machine learning models with assorted features. Although it was accurate, it's less practical because preparing all these different kinds of features and feeding them into multiple models are laborious. Recently, an end-to-end deep learning-based method [2] has shown the most accurate prediction performance in the toxicity prediction. The authors leveraged a large drug database, ZINC [74] to pre-train the network so that a modern deep learning model could be used to solve the problem. Therefore, if a deep learning model is carefully designed to effectively reflect the characteristics of the problem, we can make breakthroughs in drug discovery.

1.3 Contributions

With this motivation, this dissertation aims to design a novel deep learning based model for each of the selected sub-tasks in the drug discovery pipeline summarized in

the green boxes in Figure 1.1. In particular, we focus on the following three subtasks in this work:

- **Target identification**: We propose new feature selection methods for both disease-related and prognosis-related features. A biological target can be associated with a specific disease or a prognosis of a disease. Since the number of genes is about 20,000, we can pose the target identification problem as a feature selection task. Therefore, we introduce two feature selection algorithms with respect to disease detection and prognosis awareness.

  [**Related publications**]:

  • **Bonggun Shin***, Sungsoo Park*, Won Shim, Yoonjung Choi, Kilsoo Kang, and Keunsoo Kang, Wx: a neural network-based feature selection algorithm for transcriptomic data, Nature Scientific Report 2019 (IF=4.12)

  • **Bonggun Shin***, Sungsoo Park*, Ji Hyung Hong, Ho Jung An, Sang Hoon Chun, Kilsoo Kang, Young-Ho Ahn, Yoon Ho Ko, and Keunsoo Kang, Cascaded Wx: a novel prognosis-related feature selection framework in human lung adenocarcinoma transcriptomes, Frontiers in Genetics 2019 (IF=3.79)

- **Molecule Discovery (Drug repurposing)**: We propose a new drug-target interaction model to perform the drug re-purposing task. In this model, we present a new molecule representation to overcome the limitation of the current models. In addition to experiments on benchmarks, we also present molecule candidates from commercially available antiviral drugs that may cure the novel coronavirus (COVID-19).

  [**Related publications**]:

  • **Bonggun Shin**, Sungsoo Park, Keunsoo Kang, and Joyce C. Ho, Self-

* indicates equal contribution

Attention Based Molecule Representation for Predicting Drug-Target Interaction, MLHC 2019 (30.9%)

• Bo Ram Beck, **Bonggun Shin**, Yoonjung Choi, Sungsoo Park, and Keunsoo Kang, Predicting commercially available antiviral drugs that may act on the novel coronavirus (COVID-19) through a drug-target interaction deep learning model, Computational and Structural Biotechnology Journal (IF=4.72)

- **Molecule Discovery (Optimized drug candidate generation)**: We propose a novel drug candidate generation model that can modify an existing drug to meet given molecule properties. This model considers two subtasks at the same times: molecule generation and molecule optimization.

  [**Related publication**]:

  • **Bonggun Shin**, Sungsoo Park, and Joyce C. Ho, Controlled Molecule Generation via Self-Attention based Translation, Submitted to KDD (20%)

For each project, we present an empirical evaluation to show the competency of the proposed approaches. In addition, we also provide analyses or case studies to demonstrate the practicality of our approaches.

1.4 Outline

The rest of this dissertation is structured as three primary chapters. Chapter 2 and Chapter 3 introduce two feature selection methods one for disease-related and another for prognosis-related features. Both of them are based on neural networks and the latter is built upon the former. In Chapter 4, we present a new drug-target interaction model based on self-attention mechanism, which has shown its potential in the natural language process domain. Chapter 5 propose a new controlled molecule generating model which optimizes the input molecule to improve multiple properties in a single

end-to-end model. Finally, Chapter 6 presents the conclusion of the dissertation and discusses the future direction of the work.

2

Disease-Related Target Identification

The target identification projects primarily consist of two parts: the disease-related feature selection algorithm (Section 2) and the prognosis-related feature selection algorithm (Section 3). The two parts have been finished and published as first authored papers to Nature Scientific Report [118] and Frontiers in Genetics [142], respectively. Both of them, I was an equally contributed first author with Park.

2.1 Motivation

Gene (or protein) biomarkers clarify the health state of a patient and predict the potential response to a candidate drug as well [109]. The conventional discovery process of these biomarkers demands comprehensive requirements; a broad set of expert groups, such as biologists, statisticians, and clinicians and other experimental supports, including laboratory and complicated software tools along with another group of experts in these areas. The first stage of this process is based on the manual selection of genes or proteins with limited and biased information from the literature or experts. Then, experimental validation is followed to confirm the selected biomarker

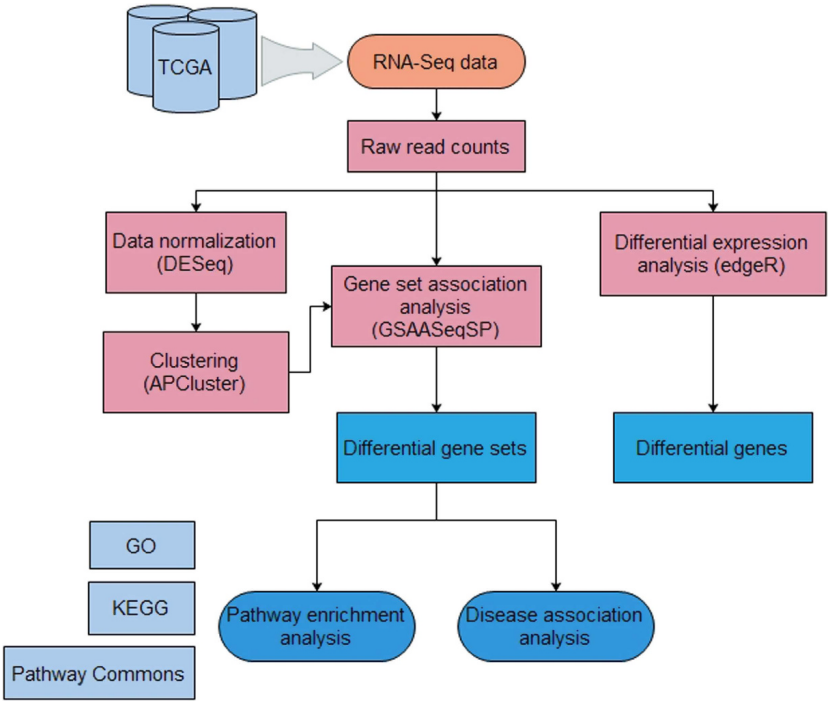


Figure 2.1: The conventional biomarker discovery pipeline using assorted machine learning methods. Figure is from [120].

candidates. The selection of a promising biomarker is important and critical because an experimental validation is costly. To satisfy these requirements, many machine learning based biomarker selection methods have been proposed such as unsupervised clustering [126], gene ontology (GO) analysis [40], sparse regularization [185], differentially expression gene (DEG) analysis [98, 130], and a hybrid of existing methods (PENG) [120].

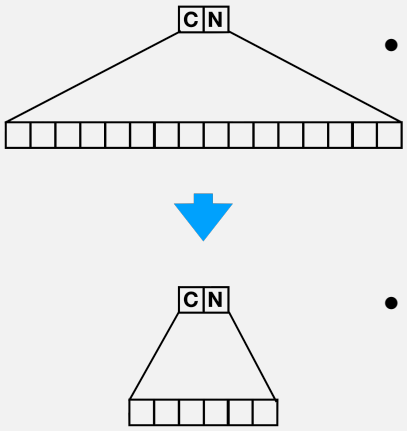
Among these methods, DEG is widely used for the identification of biomarker candidates. It narrows down to a succinct subset of genes from around 200,000 genes by looking at significantly altered expressions among different groups with a statistical threshold, an adjusted p-value of 0.05. Although it provides statistically meaningful genes, it becomes less practical recently because the number of selected genes tends to be increased to several thousand due to the increased number of raw features. The major reason for increased features is the reduced cost of sequencing. Consequently, it has become difficult for researchers to select the parsimonious sets of biomarker

candidates from a large number of DEG results. To mitigate this problem, several machine learning-based algorithms have been proposed [120, 132, 121]. In particular, [120] extracted 14-gene signatures from multiple types of cancers using a pipeline of various existing machine learning methods, as illustrated in Figure 2.1. These concise signature gene panels are valuable in the diagnosis and treatment of cancer. However, the complicated structure of this approach hinders it from widespread usage because it requires running each pipeline methods separately, which is laborious and time-consuming. To mitigate this problem, we aim to propose a new approach, called Wx, that can achieve both high predictive accuracy and simplicity for users.

Wx is our feature selection model for disease-related target identification model. The name Wx is the combination of a typical neural network weight (W) and input (x) as the feature importance is calculated based on weights and inputs together. Section 2.2 explains the formulation of the problem, and Section 2.3 details how we train the weights, and presents the feature selection algorithms, Wx.

2.2 Problem Definition

Disease-related Target Identification Task



- **Data**: Gene data with disease diagnosis
  - **X**: 20,000 different genes (features)
  - **Y**: cancer/normal (binary classification)
- **Task**: Select top k features that contributes most in the classification task.

In this project, the goal is to select a concise subset of most important genes from the whole 20,000 genes, the result of RNA sequencing data. Although [120] successfully extracted 14 genes that show the promising predictive accuracy, it is not an end-to-end model, rather it relies on many third-party libraries, which makes it less accessible. Therefore, the proposed model should not only be easy to use, but select the most informative features.

Let X be N number of gene expressions for tumor or normal samples, then it can be formally expressed as $X = \{X_1, X_2, ..., X_N\}$. Each X_i has J number of features ($X_i \in \Re^J$), each of which conveys information regarding the total expression amount of the corresponding gene. The output value $Y \in \Re^K$ is a one-hot vector that consists of K numbers depending on how many classes it represents. In formal notation, the vector Y can be expressed as $Y = [y_1, y_2, \cdots, y_K]$. A binary class case, for example, is to classify tumor samples out of normal samples. Then the i-th input data with gene expression becomes $X_i = [x_{i1}, x_{i2}, \cdots, x_{iJ}]$, and the output becomes y_i. If the i-th data is from a normal sample, then $y_i = [1, 0]$, otherwise $y_i = [0, 1]$.

2.3 Proposed Model: Wx

The proposed approach consists of three steps, training the network, extracting useful vectors, and discriminating power analysis. Throughout these steps, we use a feed-forward neural network with the softmax activation [119]. The reason why we use softmax is that the dependent variable of this task is categorical. When illustrating the idea, we assume the number of class is two for the simplicity, although the number of categories can be arbitrary as presented in Algorithm 1.

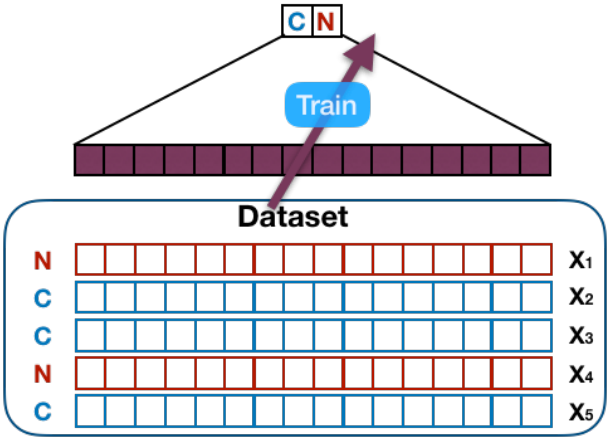


Figure 2.2: Training step of Wx method. We train the given network using the datasets consisting of genes as features and disease labels as the truth values of outputs.

[Step 1] Training the network: Figure 2.2 shows how to train the given network using the gene datasets. We denote each data sample as X_i and each corresponding annotation as either normal (N) or cancerous (C). The model is an one-layered dense network, therefore, we can denote the weights as W_N and W_C, each of these is a vector with the dimension is equal to the number of features. In general, we can express this network with the following equation:

$$\hat{Y}_{i;\Theta}(X_i) = \begin{bmatrix} P(y=[1,0,...,0]|X_i;W) \\ P(y=[0,1,0,...,0]|X_i;W) \\ \vdots \\ P(y=[0,0,...,0,1]|X_i;W) \end{bmatrix} \tag{2.1}$$

$$= \frac{1}{\sum_{k=1}^{K} \exp(\theta_k^\top X_i)} \begin{bmatrix} \exp(W_1^\top X_i) \\ \exp(W_2^\top X_i) \\ \vdots \\ \exp(W_K^\top X_i) \end{bmatrix}$$

This softmax classification network includes the model parameters $W = \{W_1, W_2, ..., W_K\}$ that are learned from the training data. With these parameters, the prediction of the output, $\hat{Y}_i$, can be expressed as Equation 2.1 along with the input X_i. This network is trained using the dataset (cancerous and normal samples as shown in Figure 2.2).

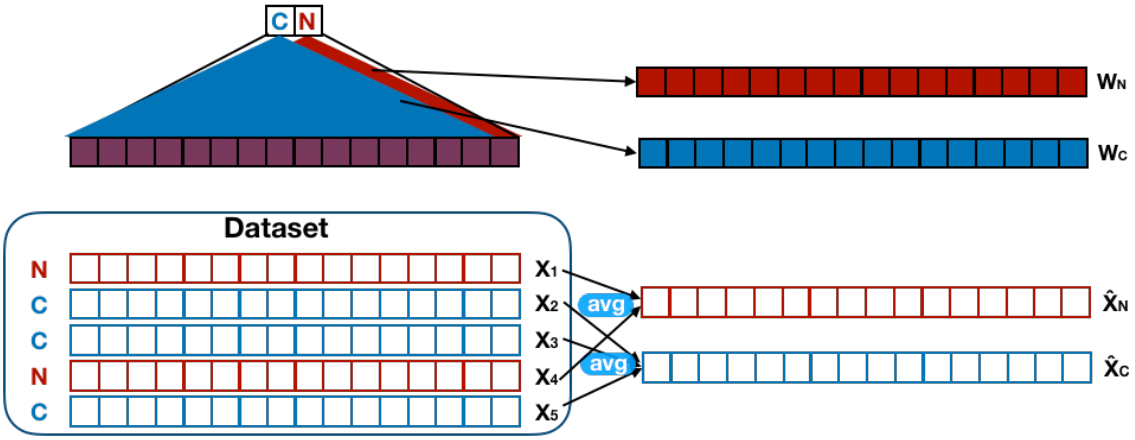


Figure 2.3: Vector extraction step of Wx method. We extract weights (W_N, W_C) and averaged input vectors $(\hat{X}_N, \hat{X}_C)$ from the trained model.

[Step 2] Extracting useful vectors: With the trained model, we extract two kinds of vectors to calculate the feature importance. The first group of vectors is the trained parameters, W. For the binary case example in Figure 2.3, the parameter vectors are W_N and W_C, the weights of normal class and the weights of cancer class, correspondingly. Another group of vectors is the averaged input vectors denoted as

$\hat{X}_N$ and $\hat{X}_C$ in Figure 2.3, which can be expressed in the following equation.

$$\hat{X}_k = \frac{1}{N_k} \sum_{X_i \in \text{Class k}} X_i \tag{2.2}$$

The summation term in Equation 2.2, $X_i \in$ Class k, represents all X_i's where $Y_i = k$. Each averaged input vector is the element-wise mean of all input vectors that belong to the class of interest. For the binary example in Figure 2.3, $\hat{X}_N$ is calculated from all normal samples, and $\hat{X}_C$ is calculated from all cancer samples.

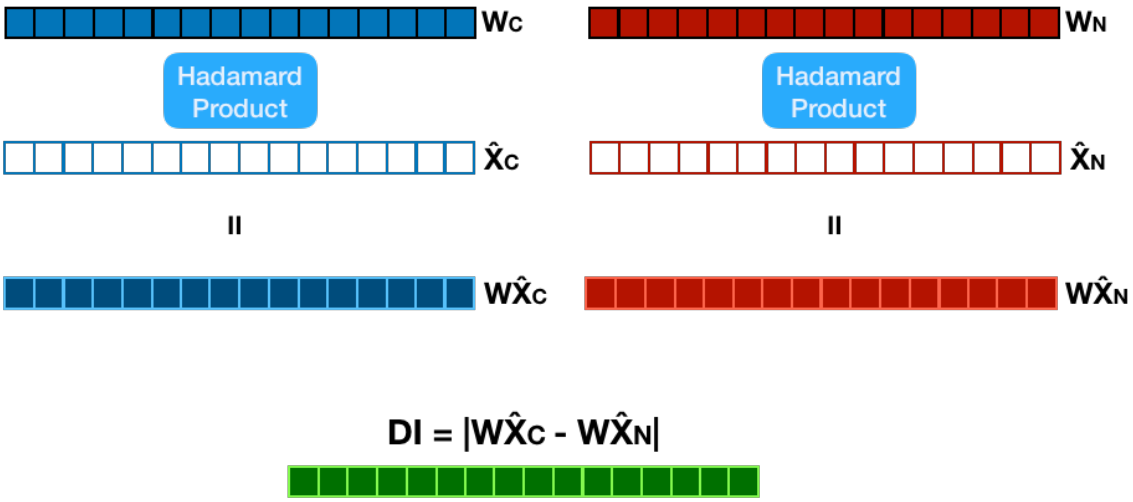


Figure 2.4: Discriminating power analysis.

[Step 3] Discriminating power analysis: For each class, we calculate an element-wise product (Hadamard product) between the corresponding weight vector and the averaged input vector as follows:

$$WX_k = W_k \odot \hat{X}_k \tag{2.3}$$

Then, we get the absolute difference from all pairs of WX_k and WX_j, which is the discriminating index (DI), the output of the method. Using this index, we can select top-k important features. For the binary example in Figure 2.3, DI is calculated from one absolute difference term from WX_C and WX_N.

Algorithm 1 generalizes all these steps for a general case, where the number of classes is greater than 2. The network parameters, W, and the dataset, X and Y, and the desired number of selected features, c, serves as the input to Discriminative Index algorithm (Algorithm 1).

Algorithm 1: Discriminative Index

input : X, Y, Θ, c
output: c number of gene names

Let X^k be the input vector with class label k;
for $k \leftarrow 1$ **to** K **do**
    $\hat{X}^k \leftarrow average(X^k)$;
    $WX_k \leftarrow \theta_k \odot \hat{X}^k$;
end
for $j \leftarrow 1$ **to** J **do**
    $DI_j \leftarrow 0$;
    foreach *combination pair* (a, b) *in* $\{1, 2, \cdots, K\}$ **do**
        $DI_j \leftarrow DI_j + |WX_a(j) - WX_b(j)|$
    end
end
$DI_{sort} \leftarrow$ *Sorted DI in descending order*;
Return top c gene names in DI_{sort};

The commentary of this algorithm is as follows.

- Classify X into K classes according to its corresponding Y which is denoted as $X^1, X^2, \ldots, X^K$.
- For each X^k, take average for all instances to form an average vector, $\hat{X}^k \in \Re^J$
- Calculate the hadamard product between the parameter related to the k'th softmax output value, θ^k and the average vector, $\hat{X}^k$, which is assigned to WX_k.
- Aggregate the element wise differences between all combination pairs of WX_k to get the discriminant index, $DI \in \Re^J$. The example with $K = 3$ is illustrated in Figure 2.4.

- After the iteration (Line 6-11 in Algorithm 1), the resulting DI is a vector of size J. This vector is sorted to form the sorted index, DI_{sort}.

- The final c features (genes) are the indices of the top c indices in DI_{sort}.

2.4 Experiments

In the experiments, we use gene expression data (mRNASeq) of 12 different cancer types from the cancer genome atlas (TCGA). Each sample contains normalized expression levels of 20,501 genes (features). A description of the TCGA data can be found in Table 2.1. We also used other RNA-seq datasets for validation of the proposed method; GSE720568 contains normalized expression levels of 23,686 genes performed in 1,257 malignant and 3,256 benign samples. GSE4041921 consists of normalized expression levels of 36,741 genes performed in 164 samples (87 lung cancer and 77 adjacent normal tissues). GSE103322 contains normalized expression levels of 23,686 genes, performed in 5,578 head and neck squamous cell carcinoma single cells (2,215 cancer cells and 3,363 non-cancer cells).

We choose the classification model as XGBoost [23] because it's known as one of the best classification models. We excluded each set of training samples when validating the classification performance. For a fair comparison, we selected top 14 features among 20,501 like [120] did in their research. In formal notation, these new inputs can be represented as $X^{v_{selected}} \in \Re^{N \times c}$. Since this dataset is imbalanced and data hungry, we use Leave-one-out cross validation (LOOCV), where only one sample is set aside for the test and the rest of them are used as the train set.

The first part of the experiments is on TCGA datasets. As summarized in Table 2.2, we compared our 14 features (Wx-14-UGCB) against Peng's 14 [120] (Peng-14-UGCB) and edgeR [130] for 12 different cancer types. Wx-14-UGCB outperformed others on average by about 1.8%p. Specifically, out of 12 cancer types, Wx-14-UGCB

Type ID	Full name	# of cancer samples	# of normal samples	# of total samples	Cancer Ratio
BLCA	Bladder urothelial carcinoma	408	19	427	0.95
BRCA	Breast invasive carcinoma	1101	113	1214	0.90
COAD	Colon adenocarcinoma	286	41	327	0.87
HNSC	Head and neck squamous cell carcinoma	522	44	566	0.92
KICH	Kidney chromophobe	65	25	90	0.72
KIRC	Kidney renal clear cell carcinoma	534	72	606	0.88
KIRP	Kidney renal papillary cell carcinoma	291	32	323	0.90
LIHC	Liver hepatocellular carcinoma	374	50	424	0.88
LUAD	Lung adenocarcinoma	517	59	576	0.89
LUSC	Lung squamous cell carcinoma	502	51	553	0.90
PRAD	Prostate adenocarcinoma	497	52	549	0.90
THCA	Thyroid carcinoma	512	59	571	0.89

Table 2.1: The number of cancer and normal samples used in this study.

Model	Wx	Peng's	edgeR
BLCA	95.79	**97.20**	94.86
BRCA	**98.19**	96.38	91.78
COAD	94.51	87.20	**98.78**
HNSC	**97.17**	92.23	94.35
KICH	95.65	95.65	**100.00**
KIRC	**99.67**	96.70	99.34
KIRP	**99.38**	97.53	**99.38**
LIHC	90.57	**94.81**	87.74
LUAD	97.92	97.58	**98.96**
LUSC	98.19	96.75	**99.28**
PRAD	**93.45**	94.55	92.36
THCA	**95.80**	89.86	90.21
Average	**96.72**	94.93	94.81

Table 2.2: Classification accuracy comparison in different cancer types of TCGA datasets(%).

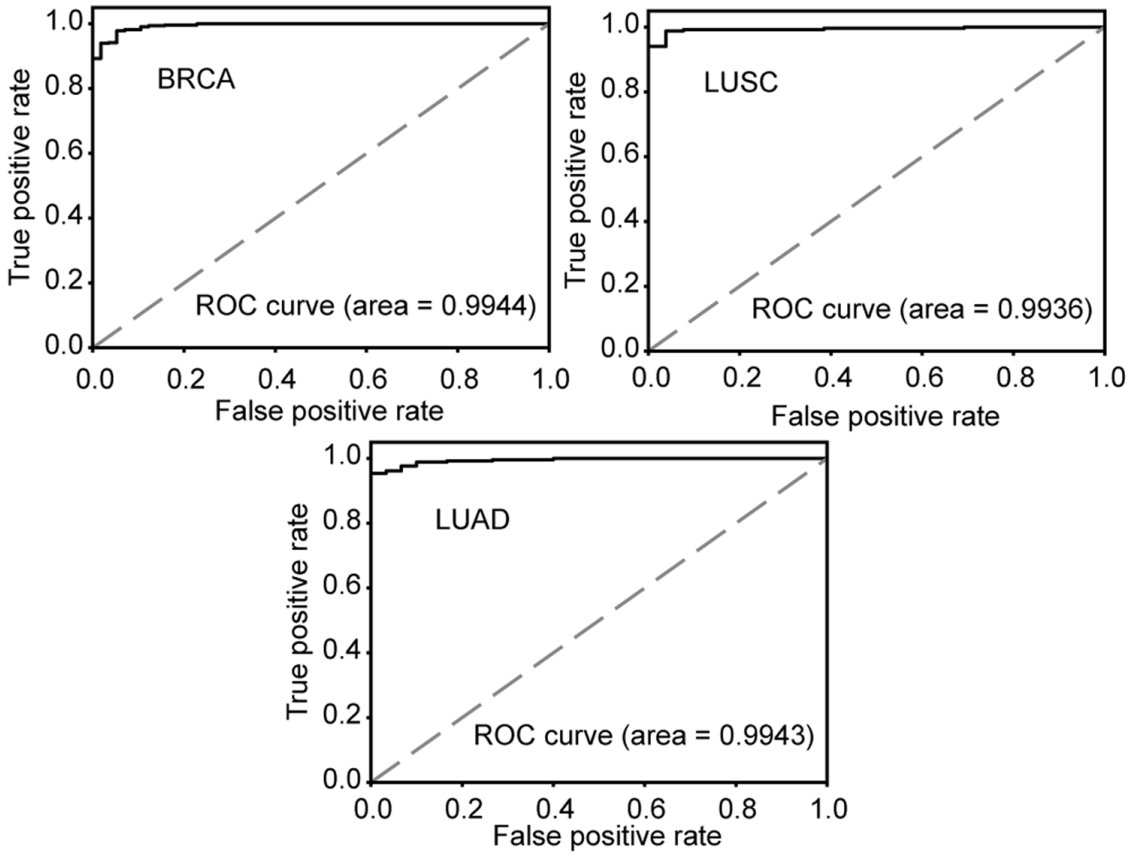


Figure 2.5: Performance of the Wx-14-UGCB on BRCA, LUAD, and LUSC RNA-seq data. AUC values are listed. ROC, receiver operating characteristic.

was the best on 6 cancer types, while Peng-14-UGCB and edgeR outperformed on two and five sets, respectively. The comparison result of area under the curve (AUC) values for BRCA, LUAD, and LUSC were 0.9944, 0.9943, and 0.9936, respectively (Figure 2.5), showing excellent classification performance of the Wx-14-UGCB set.

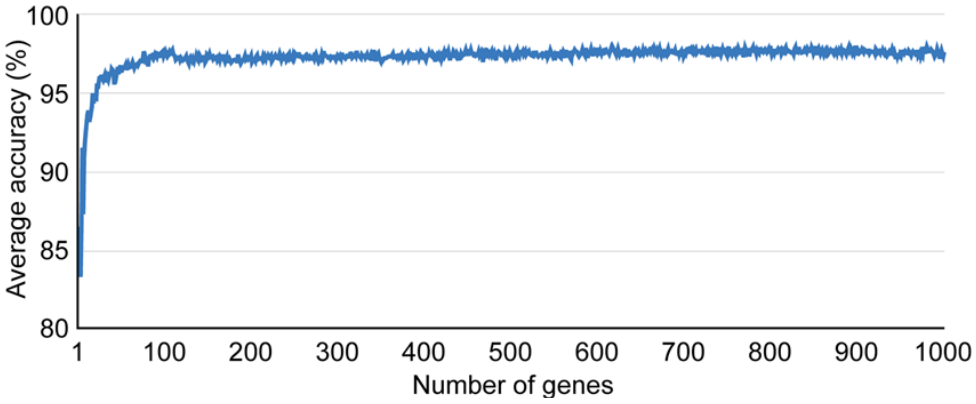


Figure 2.6: Classification accuracy according to given number of genes. The x-axis indicates the number of top genes (sorted in descending order by the DI values) used for the calculation and the y-axis represents the average accuracy.

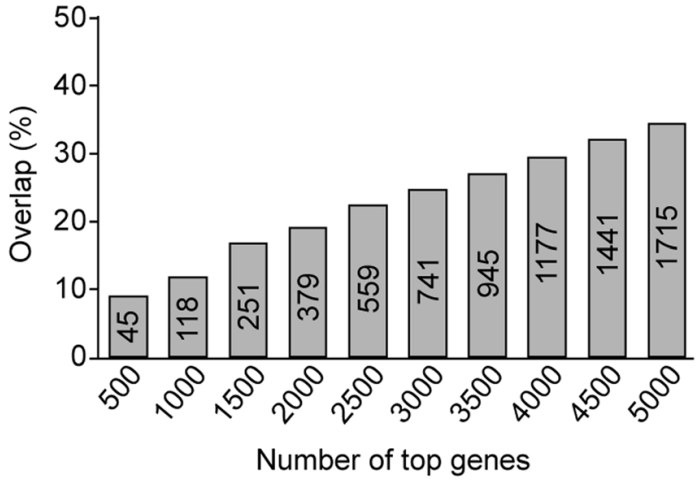


Figure 2.7: Comparison of genes identified by Wx and edgeR. The x-axis indicates the number of top genes used for the comparison and the y-axis represents the percentage of overlap between the gene sets.

As shown in Figure 2.6, approximately the top 100 UGCBs (Wx-100-UGCB) reached a plateau with the highest classification accuracy. We wondered how many genes identified by the Wx algorithm coincided with DEGs identified using edgeR.

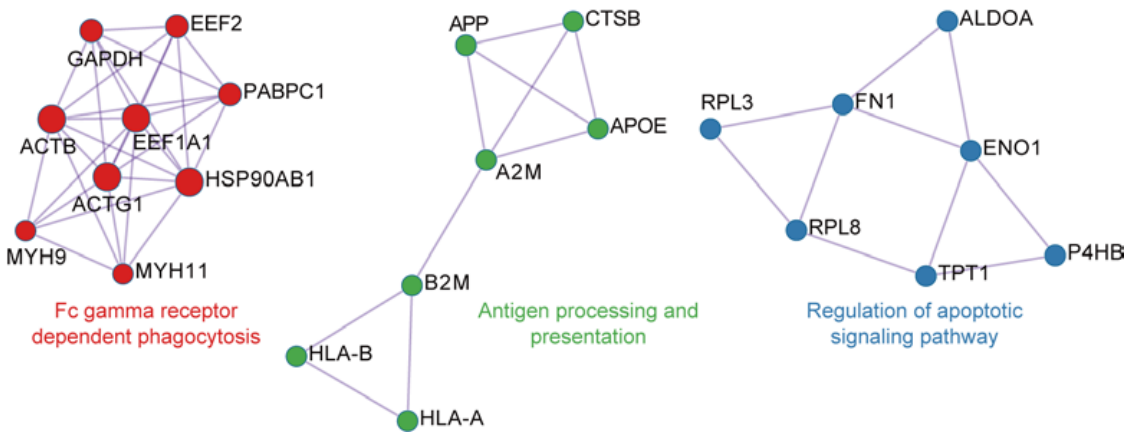


Figure 2.8: Key networks of the top 50 genes with highest DI scores are shown.

Intriguingly, less than 35% of genes overlapped (Figure 2.7). For example, a comparison of the top 500 biomarker candidate genes identified by both algorithms showed that only 45 genes (9.0%) were common. In the case of top 2,000 genes, only 379 genes (19.0%) overlapped. Thus, there was substantial discrepancy between the algorithms with the same gene expression data. Next, we performed gene ontology (GO) and network analysis to investigate the putative function of top 50 UGCBs using Metascape [156]. Genes involved in the Fc gamma receptor dependent phagocytosis, antigen processing and presentation, and regulation of apoptotic signaling pathway were significantly altered (Figure 2.8), suggesting that the deregulation of these pathways might be a critical factor in the onset or progression of most cancers. Further investigations of these genes are warranted.

GSE id	Cancer type	Wx-14-UGCB	Peng-14-UGCB
GSE72056	Melanoma	**90.71**	70.22
GSE40419	Lung adenocarcinoma	**80.00**	56.87
GSE103322	Head and neck squamous cell carcinoma	**81.10**	68.28

Table 2.3: The classification accuracy of the UGCBs for non TCGA dataset (%).

We further validated the performance by evaluating the classification accuracy of Wx-14-UGCB and Peng-14-UGCB with cancer and normal RNA-seq data from three

independent cancer studies including a melanoma (GSE72056), Lung adenocarcinoma (GSE40419), and head and neck squamous cell carcinoma (GSE103322) that had not been included in the 12 types of TCGA cancer cohort [179, 154]. We calculated the classification accuracy by dividing the samples in a given cohort into the training set (2,888 samples, 64%), validation set (723 samples, 16%), and test set (902 samples, 20%). Then, the training set was used to train a model using a neural network (NN) algorithm and the validation set was used to assess how well the model had been trained. Finally, the test set was used to calculate the classification accuracy with the trained model. As shown in Table 2.3, Wx-14-UGCB significantly outperformed Peng-14-UGCB by large margin. Specifically, with the expression levels of the genes in the Wx-14-UGCB set, 818 out of the 902 test samples were correctly classified, whereas 633 out of 902 test samples were correctly classified using the Peng-14-UGCB set. For the lung adenocarcinoma data set (GSE40419), Wx-14-UGCB showed 80.00% classification accuracy when classifying lung cancer and adjacent normal cells, while Peng-14-UGCB showed 56.87% classification accuracy. For the 5,578 head and neck squamous cell carcinoma single cells (2,215 cancer cells and 3,363 non-cancer cells) (GSE103322), Wx-14-UGCB showed 81.10% classification accuracy when classifying cancer and non-cancer cells, while Peng-14-UGCB showed 68.28% classification accuracy. In summary, the top 14 genes (Wx-14-UGCB) identified by the Wx algorithm could potentially be used as novel gene expression biomarkers for the detection of various types of cancers, although its use might be limited by clinical difficulties associated with RNA-based applications. Further experimental investigations are required to validate the Wx-14-UGCB.

2.5 Discussion

The proposed algorithm estimates the classification power of genes in a given gene expression data set using the discriminative index (DI) score algorithm. Researchers can intuitively select gene-expression biomarker candidates from the DI scored gene list.

Cancer type	Wx-14-UGCB	Peng-14-UGCB
BLCA BRCA COAD(READ) HNSC KICH KIRC KIRP LIHC LUAD LUSC PRAD THCA	EEF1A1, FN1, GAPDH, SFTPC, AHNAK, KLK3, UMOD, CTSB, COL1A1, GPX3, GNAS, ATP1A1, SFTPB, ACTB	KIF4A, NUSAP1, HJURP, NEK2, FANCI, DTL, UHRF1, FEN1, IQGAP3, KIF20A, TRIM59, CENPL, C16ORF59, UBE2C

Table 2.4: Gene expression biomarkers identified by different studies.

Our interesting finding is that the 14 gene signatures (Wx-14-UGCB) identified by the Wx algorithm includes the housekeeping gene GAPDH (Table 2.4), which has been used in many studies as a control (or reference) gene. Recently, several concerns about using the GAPDH gene as a housekeeping gene has been reported [56, 46, 6, 146, 17]. Our result also indicated that the GAPDH gene was one of the highest DI-scored genes, and this gene should therefore be used with caution as a control gene in gene expression experiments. Interestingly, another well-known housekeeping gene ACTB was ranked 14 out of 20,501 genes (Table 2.5), suggesting that both GAPDH and ACTB genes might be unsuitable housekeeping genes for gene expression experiments, particularly in cancer studies. Further investigations of the remaining genes such as FN1, EEF1A1, COL1A1, SFTPB, SFTPC, and ATP1A1 will shed light on the identification of novel biomarker genes for a pan-cancer cohort.

One of the disadvantages of artificial neural network-based approaches when ap-

Rank	Gene	Discriminative Index (arbitrary number, higher is better)
1	EEF1A1	1.65849
2	FN1	1.61224
3	GAPDH*	1.50260
4	SFTPC	0.96081
5	AHNAK	0.71224
6	KLK3	0.56306
7	UMOD	0.55580
8	CTSB	0.42822
9	COL1A1	0.41349
10	GPX3	0.37308
11	GNAS	0.36476
12	ATP1A1	0.34630
13	SFTPB	0.33725
14	ACTB*	0.32997
15	ACPP	0.32805
16	FTL	0.31993
17	P4HB	0.31076
18	A2M	0.30867
19	PIGR	0.29527
20	DCN	0.29410
21	EEF2	0.28864
22	CLU	0.28477
23	ACTG1	0.25872
24	PABPC1	0.24866
25	SPARC	0.24861
26	CTSD	0.24328
27	RPL3	0.23105
28	RPL8	0.22458
29	ALDOA	0.21630
30	B2M	0.21391
31	MYH11	0.21365
32	TPT1	0.20991
33	HLA-B	0.20859
34	TXNIP	0.20725
35	HSP90AB1	0.20676
36	MGP	0.20396
37	APP	0.20064
38	PKM2	0.19627
39	ALB	0.19292
40	ALDOB	0.19162
41	KRT13	0.18497
42	C4A	0.18036
43	CALR	0.17827
44	APLP2	0.17746
45	ENO1	0.17689
46	HLA-A	0.17441
47	GSN	0.17034
48	COL1A2	0.16909
49	MYH9	0.16713
50	APOE	0.16215

Table 2.5: Genes ranked by the discriminative index score. * indicates genes known as house keeping genes.

Number of top features (genes) identified by WX	Classifier	Liver (GSE105127) Pericentral (n=19)	Intermediate (n=19)	Periportal (n=19)
10	XGBoost	57.89	50.00	50.00
	SVM	68.42	72.22	0.00
20	XGBoost	78.95	66.67	50.00
	SVM	73.68	44.44	0.00
30	XGBoost	84.21	72.22	62.50
	SVM	73.68	27.78	6.25
50	XGBoost	84.21	66.67	50.00
	SVM	78.95	33.33	0.00
100	XGBoost	84.21	61.11	62.50
	SVM	84.21	55.56	0.00
200	XGBoost	84.21	77.78	81.25
	SVM	84.21	61.11	0.00

Table 2.6: Classification accuracy (%) of non-cancer transcriptomic data set.

plied to biomedical data is that a large number of samples are needed to achieve good classification or regression performance. We observed relatively lower classification accuracy (Table 2.6) when the Wx algorithm was applied to a non-cancer transcriptomic data set (GSE105127), which contains normalized expression levels of 65,671 transcripts performed in the pericentral ($n = 19$), intermediate ($n = 19$), and periportal ($n = 19$) regions of the human liver isolated by laser-captured microdissection [14]. In addition, selected features from the same data set vary depending on algorithms. In our comparison, there were no overlaps between the top 14 genes identified by Wx or Pengs. This kind of inconsistency is caused mainly by the algorithmic difference, as reported in several differentially expressed gene analysis studies [50, 29, 164]. Thus, it is difficult to establish which algorithm is better by comparison without experimental verification. Therefore, the usefulness of the 14 genes (Wx-14-UGCB) for cancer biomarkers should be validated with extensive experimental evidence in the near future.

In summary, the Wx algorithm developed in this study estimates the classification

power of genes in a given gene expression data set using the discriminative index (DI) score algorithm. Researchers can intuitively select gene-expression biomarker candidates from the DI scored gene list. Further experimental validation will be necessary to prove the Wx algorithms usefulness.

2.6 Contribution

In this project, my contributions are as follows.

- Designed the study with Park and Kang.
- Developed the algorithm with Park and Shim.
- Analyzed the result with Ki.Kang, Park, Ke.Kang, and Choi.
- Wrote the manuscript with Ki.Kang, Park, Ke.Kang, Shim, and Choi.

3

Prognosis-Related Target Identification

In the second biomarker identification project, we aim to extend the first feature selection method (Chapter 2) to a regression task. The first feature selection method (Chapter 2) is dealing with a discrete target label, while the target value of this project is a survival time, which is continuous. Therefore, we cannot directly use the previous method in a prognosis-related feature selection problem. To solve this problem, we present a new feature selection method called Cascaded Wx algorithm.

3.1 Motivation

In this project, the goal is to select a concise subset of most important genes from the whole 20,000 genes, the result of RNA sequencing data that can classifying high/low risk cohorts. We focus on lung caner patients because lung cancer is the most commonly diagnosed cancer and the second most common cause of cancer-related deaths worldwide [12]. Most lung cancer cases are nonsmall cell lung cancer (NSCLC), and lung adenocarcinoma (LUAD) accounts for more than 50% of all NSCLCs. Recently, survival rates for LUAD patients have been greatly improved with the development

of improved treatment approaches, including surgical or radiation techniques, and the introduction of targeted therapies and immunotherapies tailored to the molecular or immunologic characteristics of tumors. However, the survival rate is still only about 50% for potentially curatively resected LUAD [171]. To optimize clinical intervention, it is important to identify which patients have poor prognoses. The prediction of prognosis requires an extensive knowledge of various aspects of cancer biology and an understanding of relevant clinical information such as TNM stage, histology, and genetic mutations [61]. Among the clinical features, TNM staging is the most successful clinical parameter in practice and is widely used to predict patients prognoses. However, this staging method still has room for improvement in the era of genomic sequencing, where abnormalities in multiple genes can be detected simultaneously [134]. Among the various genome-wide applications, the gene expression signature is the most promising approach to the prediction of clinical outcomes [159, 128, 24], as a suite of expressed genes reflects the identity of a given cell population. Several gene expression-based clinical applications such as MammaPrint [168] and Oncotype DX [18] are being used in clinical practice. These applications predict patients' prognoses and drug and/or chemotherapy responsiveness by examining the expression levels of a defined gene set. Therefore, the identification of a particular gene set associated with clinical findings is crucial in many disease research studies.

Recent technological advancements in clinical genome sequencing using next-generation sequencing (NGS) technologies provide opportunities to understand the relationships between gene expression and tumor phenotypes [86]. For example, several studies classify NSCLC patients into subgroups with differing clinical outcomes using gene signatures [22, 148, 11, 173]. However, the results of such studies have been unsatisfactory in terms of discrepancies between identified gene signatures. The possible reasons for the inconsistent results among the studies include the use of small samples compared to the number of genes (high-dimensional data), the use of differ-

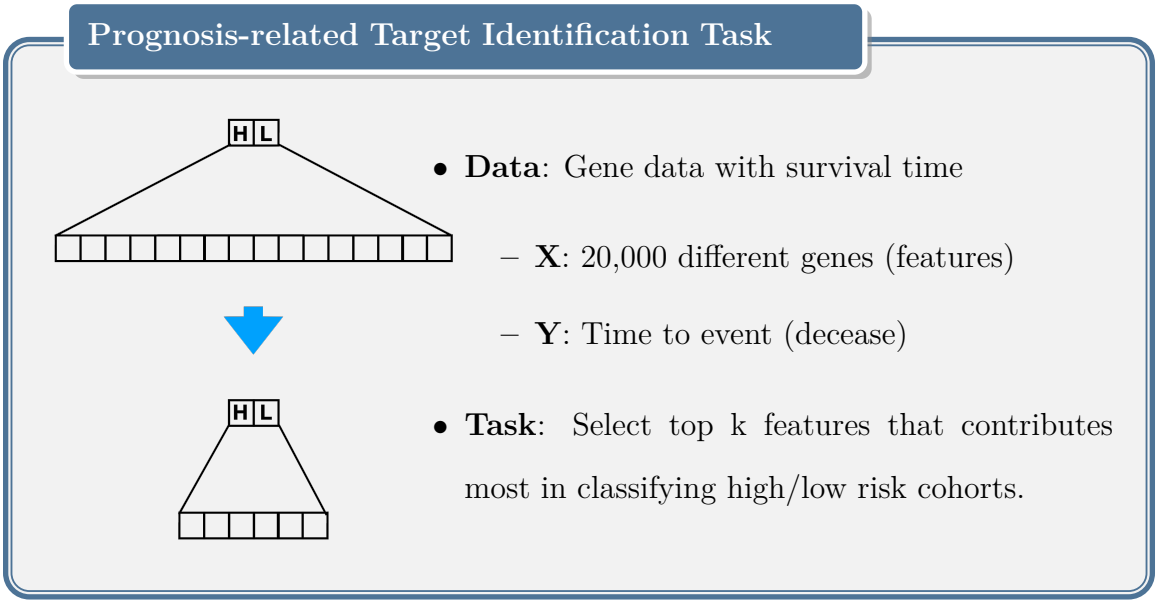
ent platforms, and the problems with feature preprocessing steps. In addition, there are no robust methods for analyzing such high- dimensional data effectively.

Machine learning (ML) algorithms can be a useful approach to the analysis of high volumes of data if a model is well constructed with high-quality input data for training. Numerous variations of the original ML algorithms have been developed and applied to a variety of problems [95, 118, 183, 49]. In molecular biology, NGS technologies, which revolutionized the profiling approach by sequencing huge numbers of given short DNA fragments, have been generating enormous amounts of data these days [60]. Because of this, there is an urgent need to develop ML-based algorithms that can effectively analyze such high volumes of genomic data. Support vector machines (SVM; [21]), k-nearest neighbors [30], multilayer perceptrons [103], decision trees [25], random forest (RF; [182]) algorithms, logistic regression, and gradient boosting machines [102] are ML algorithms that are frequently used to analyze big data. However, these methods were not originally designed to extract prognostic features from patients' data. Recently, several ML-based algorithms have been proposed to select a subset of key features (genes) for classification [3, 176, 52] or to identify prognostic features [167] from high-throughput molecular profiling data. There is still room for improvement, however, as new deep learning algorithms continue to emerge in the field of ML [41, 122].

To effectively analyze multidimensional datasets, dimension-reduction algorithms such as feature selection are often required. Principal component analysis (PCA; [169]), nonnegative matrix factorization [90], kernel PCA [107], graph-based kernel PCA, linear discriminant analysis [106], and generalized discriminant analysis [7] are algorithms that are widely applied to high-dimensional biomedical datasets. In addition to these approaches, several studies recently used artificial neural networks to predict clinical outcomes in lung cancer patients [77, 172, 64]. However, these approaches do not fully take into account available information such as high-throughput

profiling data (e.g., transcriptomes) and/or clinical information for feature selection. To address these problems, we developed a novel feature selection framework called Cascaded Wx (CWx) to enhance the efficiency of feature selection and the accuracy of prediction for given patients prognosis. Our analyses revealed that the CWx framework selected more prognosis-related features than algorithms in categories such as similarity-based, sparse learning-based, ML-based, and statistical-based models, highlighting the potential value of our proposed framework for biomedical data.

3.2 Problem Definition



Gene expression data (mRNASeq) of The Cancer Genome Atlas (TCGA) lung adenocarcinoma (LUAD) were obtained from Firehose at The Broad Institute (`https://gdac.broadinstitute.org`). From the whole data, we extracted gene features, survival values, and censoring information, which can be formally represented as $X \in \Re^{n \times d}$, $S \in \Re^{n}$, and $C \in \Re^{n}$, respectively; n is the number of patients, and d is the feature dimension. If $C_i = 0$ (uncensored patients), the survival time interval (S_i) represents the time between the start of observing the patient status and the

event time (date of death). If a patient datum is right censored ($C_i = 1$), the survival time interval (S_i) represents the time elapsed between the start of observing the patient status and the end of the study. Among the 507 LUAD patients, there are 183 uncensored (death event occurred) samples and 324 right-censored samples. Each sample contained read counts (expression levels) of 20,501 genes. These count-based values are abundant for a few specific transcripts (highly expressed genes), a factor that prevents a model from finding a good pattern. To mitigate this problem, we use a log transformation:

$$X_{ij}^{new} = \log_2(X_{ij} + 1)$$

for $i \in n$ and $j \in d$. A constant, 1, is added to the read count value of each gene before applying the logarithm function to avoid the numerical problems. Min-max normalization is then applied to the log-transformed data. With these datasets, the objective is to select the optimal gene set associated with patients prognoses using the survival information.

3.3 Proposed Model: Cascaded Wx

The proposed method was based on the Wx algorithm [118] (Chapter 2), which identifies key genes discriminating different classes. As this method was designed for a classification problem, we extend it to be applicable to the patient cohort grouping task with continuous survival values. The basic concept of the proposed algorithm (cascaded WX, CWx) is to guide the feature selection algorithm by efficiently organizing the training curriculum from an easy to hard one. For each stage we select a subset of the training samples (patients). There are multiple stages with different difficulty levels. By doing this, the model will automatically reduce the number of features (genes). The three stage example is summarized in (Figure 3.1). In the first step, patients are divided into high- and low-risk cohorts according to whether

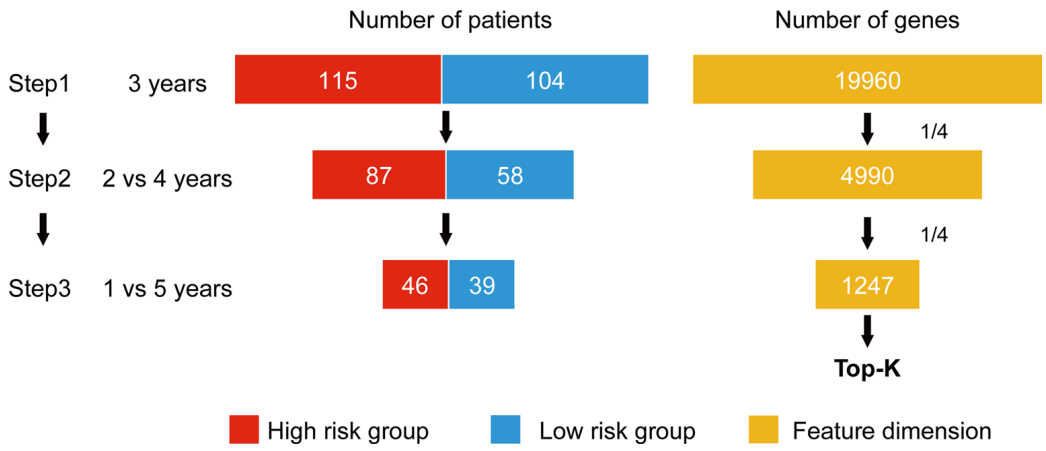


Figure 3.1: An example of CWx's feature selection procedure. Input samples (patients) are reduced through three cascaded steps using different criteria. For each step, a different cutoff criteria is used in an easy to hard manner. For example, three-year, 2 versus 4-year, and 1 versus 5-year cutoffs for grouping samples into either high- or low-risk groups are used at the first, second, and third steps, respectively. Input features (genes) are also reduced by a quarter in each step. Finally, the prognostic potential of features can be estimated according to the weights calculated from the trained neural network.

they have survived for 3 years. For example, 115 deceased patients within 3 years in a training set formed one group (28.4%; high risk), whereas 104 patients who lived more than 3 years formed another group (25.7%; low risk). The remaining patients (186, 45.9%) were right censored, meaning that there was no information as to whether these patients were deceased within 3 years. These right-censored patients are excluded in the training stage. The second and third steps are similar to the first step with different cutoffs (2 versus 4 years and 1 versus 5 years, respectively). As the number of samples is decreased by each step's criteria, the number of features (genes) was also reduced by a quarter in each step. One quarter of the features were selected according to the importance determined by our previous Wx feature selection algorithm. A total of 19,960 genes were used as input features after removing genes with no variance. The final output after these three steps is a set of genes ranked by prognostic weights, estimated in a manner similar to the Wx algorithm.

3.4 Experiments

We compared the proposed algorithm, CWx, to the following supervised feature selection algorithms from five different categories: i) ML based models: RF [13], SVM [28], and Extreme Gradient Boosting (XGBoost) [23]; ii) similarity based models: fisher score [45], ReliefF [87], and trace ratio (Trace ratio) [110]; iii) sparse learning-based models: multi-task feature learning via efficient $l2$,1-norm minimization (LLL21) [96] and robust feature selection (RFS) [111]; iv) statistical based models: Fscore; and V) others : cox proportional hazard (CoxPH) [31]. These algorithms calculate a score for each given feature, so the performance of each cancer prognosis prediction can be estimated by comparing the highest-scoring features selected by each algorithm. We also compared CWx to CoxPH and Coxnet as baseline methods for prognosis prediction. Feature selection criteria for CoxPH and Coxnet were P value and beta coefficients, respectively.

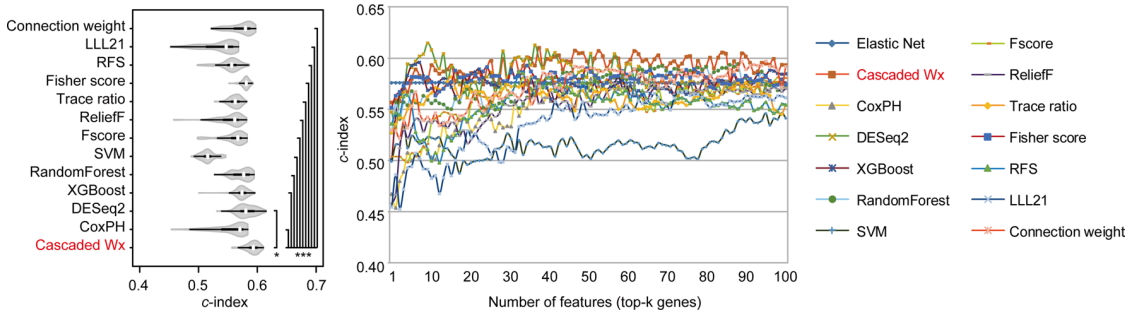


Figure 3.2: Violin plot of the comparison of feature selection algorithms with top 100 genes. The metric is a c-index with the selected 100 genes in lung adenocarcinoma (LUAD) samples for each algorithm (left). White circles indicate the medians; box limits inside the polygons indicate the 25th and 75th percentiles as determined by R software; whiskers extend 1.5 times the interquartile range from the 25th and 75th percentiles; polygons represent density estimates of data and extend to extreme values. Asterisks ($*p < 0.05$ and $***p < 0.001$) indicate the results of one-way ANOVA ($p < 0.0001$) with post hoc test (pairwise t test with BonferroniHolm correction). x- and y-axes indicate the number of cumulative top genes and c-index, respectively (right).

The results indicated that CWx was superior to the other methods in terms of

c-index when comparing the top genes (cumulative) from 1 to 100 in LUAD samples (Figure 3.2).

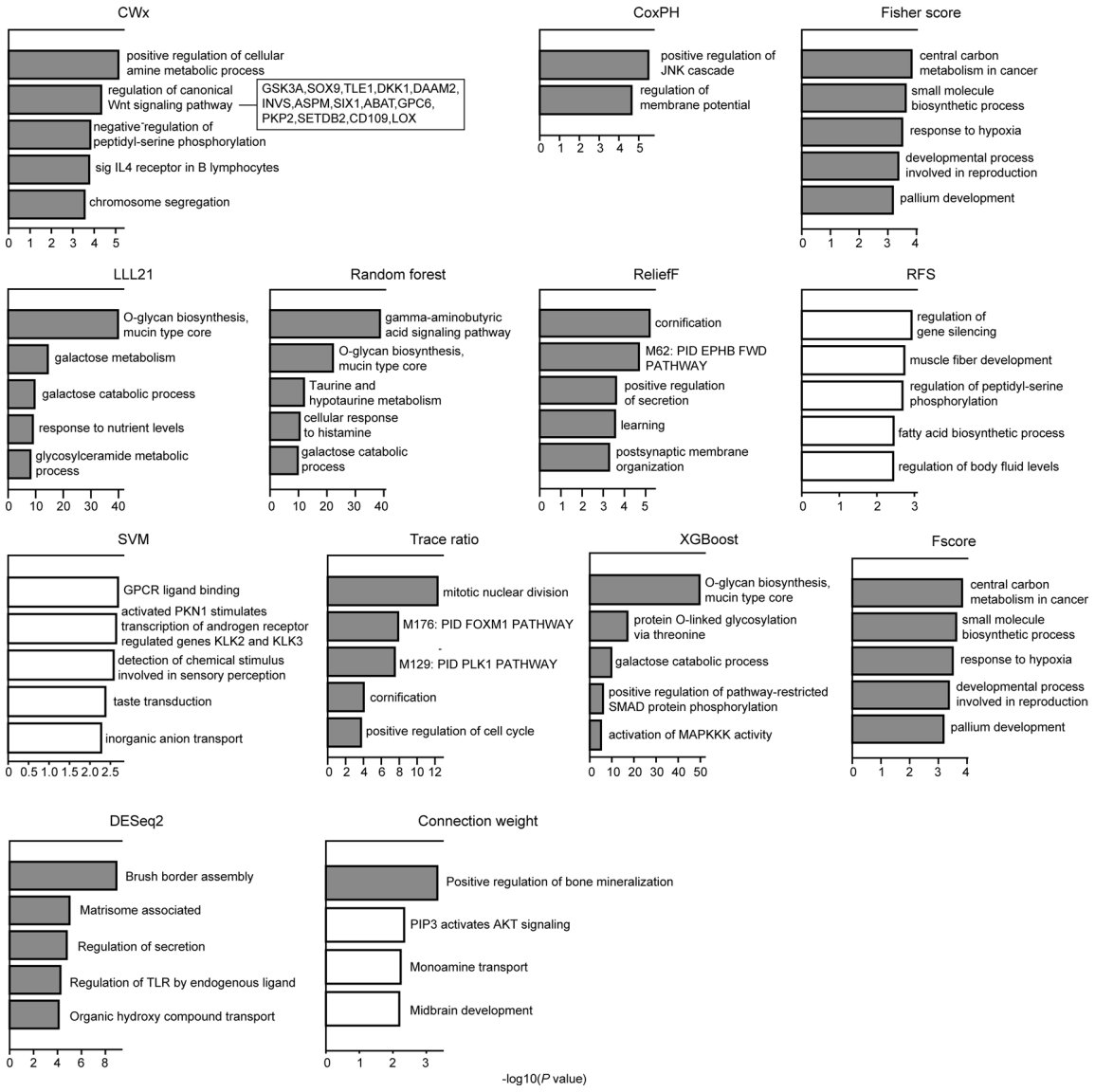


Figure 3.3: Gene ontology (GO) analysis of top 100 genes. GO analysis was performed using Metascape (http://metascape.org/gp/index.html) with top 100 genes (default parameters were used). The significance of a given GO term is represented by gray (significant) or white (nonsignificant) bars with a P cutoff value of 0.0001.

Next, we performed Gene Ontology (GO) analysis to identify the biological pathways associated with the top 100 genes. This analysis revealed that the gene set identified by CWx was associated with the Wnt signaling pathway (Figure 3.3), one of

the key pathways regulating development, and closely associated with many cancers. The gene sets identified by the other algorithms were related to different pathways such as “positive regulation of JNK cascade” (CoxPH), “central carbon metabolism in cancer” (Fisher score and Fscore), “O-glycan biosynthesis, mucin type core” (LLL21, RF, and XGBoost), “mitotic nuclear division” (Trace ratio), “regulation of gene silencing” (RFS), and “GPCR ligand binding” (SVM). Differences between the gene sets identified by the different algorithms, and their associated biological pathways, need to be further investigated in future studies.

3.5 Discussion

Our evaluation using 507 TCGA LUAD transcriptomes revealed that the proposed model outperformed the other methods by effectively training the algorithm from an easy to hard problem. This finding means that the gene set found by CWx is one of the best candidate gene sets to predict patients prognoses. In addition, CWx has a linear execution time to complete the feature selection steps depending on the number of samples. While the information theoretical-based feature selection algorithms take longer to finish the feature selection procedure.

One of the key pathways related to the prognosis of LUAD patients identified by the CWx framework was the Wnt signaling pathway. A recent study has shown that two distinct sub-populations of cells, one with high Wnt signaling activity and another forming a niche that provides the Wnt ligand, are activated in LUAD. In addition, in vitro and in vivo studies have suggested that Wnt responsiveness contributes to the survival of cancer cells and the maintenance of a stem cell-like niche cell phenotype [151]. Not only that, many CWx found genes have been previously reported in LUAD studies [161, 101, 78, 1, 139, 51, 177].

3.6 Contribution

In this project, my contributions are as follows.

- Designed the study with Park, Ko, and Kang.
- Developed the algorithm with Park.
- Wrote the manuscript with Park, Hong, Ko, and Kang.

4

Drug Repurposing

The molecule discovery is divided into two tasks: drug repurposing (Chapter 4) and optimized drug candidate generation (ODG, Chapter 5). The result of the drug repurposing project have been published to the machine learning for healthcare conference [144]. In addition, one of the case studies has been submitted to Computational and Structural Biotechnology Journal.

4.1 Introduction

Many diseases are caused by abnormal protein levels, therefore, a drug is designed to target particular proteins. However, a drug may not work well for a decent portion of patients, because an individual's response to a drug varies depending on the genetic inheritance [163]. Unfortunately, pharmaceutical companies focus only on a majority cohort of patients as drug discovery is an expensive process. The reduction of the cost of the drug discovery process will not only lead to drugs costing less, resulting in reduced healthcare costs for a patient but can also allow companies to develop personalized drugs based on genetics.

Among the many parts of the drug discovery process, predicting drug-target interactions (DTI) is an essential one. DTI is difficult and costly as experimental assays

not only take significant time but are expensive. Furthermore, only less than 10% of the proposed DTIs are accepted as new drugs [66]. Therefore, *in silico* (performed on a computer) DTI predictions are much demanded since it can expedite the drug development process by systemically suggesting a new set of candidate molecules promptly, which can save time and reduce the cost of the whole process by up to 43% [43].

In response to this demand, three types of *in silico* DTI prediction methods have been proposed in the literature: molecular docking, similarity-based, and deep learning-based models. Molecular docking [157, 100] is a simulation-based method using the 3D structured features of molecules and proteins. Although it can provide an intuitive visual interpretation, it is difficult to obtain a 3D structure of a feature and cannot scale to large datasets. To mitigate these problems, two similarity-based methods, KronRLS [116] and SimBoost [66] have been proposed using efficient machine learning methods. However, using a similarity matrix has two downsides. Firstly, feature representation is limited in the similarity space, thereby ignoring the rich information embedded in the molecule sequence. For example, if a brand new molecule is tested, the model will represent it using relatively unrelated (dissimilar) molecules, which would make the prediction inaccurate. Secondly, it necessitates the calculation of the similarity matrix which can limit the maximum number of molecules in the training process. To overcome these limitations, a deep learning-based DTI model, DeepDTA [114], was proposed. It is an end-to-end convolutional neural network (CNN)-based model that eliminates the need for feature engineering. The model automatically finds useful features from a raw molecule and protein sequence. Its success has been demonstrated on two publicly available DTI benchmarks. Although this work illustrated the potential of a deep learning-based model, there are several areas for improvement:

- CNNs can't model potential relationships among distant atoms in a raw molecule

sequence. For example, with three layers of CNNs each with a filter size of 12, the model can capture associations in atoms up to 35 distances in a sequence. We posit that the recently proposed self-attention mechanism can be used to capture any relationship among atoms in a sequence, and thereby provide a better molecule relationship

- The one-hot encoding used to represent each molecule fails to take advantage of existing chemical structure knowledge. An abundance of chemical compounds are available in the PubChem database [55], from which we can extract useful chemical structures for pre-training the molecule representation network.

- Fine-tuning is a type of transfer learning where weights trained from one network can be transferred to another so that the weights can be adjusted to the new dataset. Thus, we can transfer the weights learned from the PubChem database to our DTI model. This will help our model to use the learned knowledge of a chemical structure while tailoring it to predicting DTI interactions.

With these observations, we propose a new deep DTI model, Molecule Transformer DTI (MT-DTI), based on a new molecule representation. We use a self-attention mechanism to learn the high-dimensional structure of a molecule from a given raw sequence. Our self-attention mechanism, Molecular Transformer (MT), is pre-trained on publicly available chemical compounds (PubChem database) to learn the complex structure of a molecule. This pre-training is important, because most datasets available for DTI training has only 2000 molecules, while the data for pre-training (PubChem database) contains 97 *millions* of molecules. Although it does not contain interaction data but just molecules, our MT is able to learn a chemical structure from it, which will be effectively utilized when transferred to MT-DTI (our model). Therefore, we transfer this trained molecule representation to our DTI model so that it can be fine-tuned with a DTI dataset. The proposed DTI model is evaluated on two

well-known benchmark DTI datasets, Kiba [152] and Davis [35], and outperforms the current state of the art (SOTA) model by 4.9% points for Kiba and 1.6% points for Davis in terms of area under the precision-recall curve. Additionally, we demonstrate the usefulness of our trained model using a known drug list targeting a specific protein. The trained model generates all FDA approved drugs with high rankings in the drug candidate lists. The demonstrated effectiveness of the proposed model can help reduce the cost of drug discovery. Furthermore, precise molecule representation can enable drugs to be designed for specific genotypes and potentially enable personalized medicine.

Technical Significance We propose a novel molecule representation, adapting the self-attention mechanism that was recently proposed in Natural Language Process (NLP) literature. This is inspired by the idea that understanding a molecule sequence for a chemist is analogous to understanding a language for a person. We introduce a new way to train the molecule representation model to fit the DTI problem using an existing corpus to achieve a more robust representation. With this (pre)trained molecule representation, we fine-tune the proposed DTI model and achieve new SOTA performances on two public DTI benchmarks.[1]

Clinical Relevance With our new model, we can potentially lower medication costs for patients, which can help make drugs more affordable and help patients be more adherent. In addition, this can serve as the stepping stone for designing personalized medication. Through the proper representation of molecules and proteins, we can better understand the properties of patients that make a drug helpful or not [127].

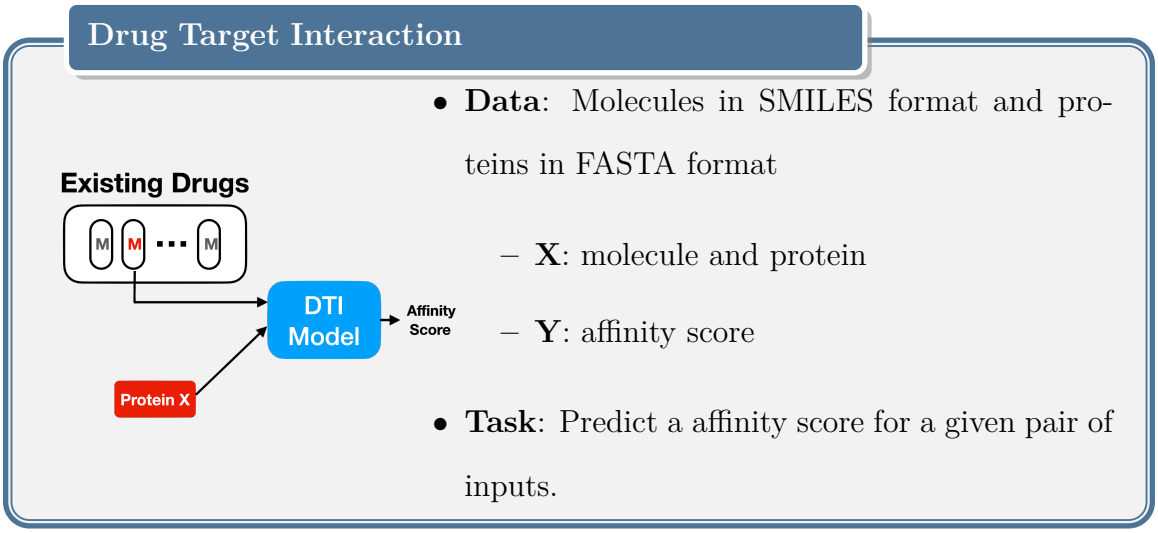
[1]The demo is publicly available at
: `https://mt-dti.deargendev.me/`

4.2 Related Work

Predicting drug-target interaction traditionally focused on a binary classification problem [174, 10, 158, 15, 58, 27, 16, 113]. The most recent approach tackling this binary classification problem is an interpretable deep learning based model [53]. Although these methods show promising results on binary datasets, they are simplifying protein-ligand interactions by thresholding affinity values. In order to model these complex interactions, several methods have been proposed, which can be categorized into three kinds. The first category of these models is molecular docking [157, 100], which is a simulation-based method. These methods are not scalable, due to heavy preprocessing. To overcome this downside, the second category, similarity-based methods, was proposed. They are KronRLS [116] and SimBoost [66], which is based on the calculation of similarity matrix of inputs. With the advent of deep learning, two deep learning-based methods have been proposed [53, 114]. Like these models, our model is also based on deep learning, but our proposed model has a better molecule representation, and improves its performance through a transfer learning technique.

Deep learning-based transfer learning, pre-training and fine-tuning, have been applied to various tasks such as computer vision [133, 54], NLP [71], speech recognition [75, 99], and health-care applications [141]. The idea is to use appropriate pre-trained weights to improve results in corresponding tasks, which also can be found in our experimental results.

4.3 Problem Definition



4.4 Proposed Model: Molecule Transformer-Drug Target Interaction

We introduce a new drug-target interaction (DTI) model and a new molecule representation in this section. The basic motivation of the proposed model is that the structure of molecule sequences is shown to be very similar to the structure of natural language sentences in that contextual and structural information of atoms are important when understanding the characteristics of a molecule [76]. Specifically, each atom interacts with not only neighboring atoms but also long distant ones in a simplified molecular-input line-entry system (SMILES) sequence, a notation that encodes the molecular structure of chemicals. However, the current SOTA method using CNNs can't relate long distance atoms when representing a molecule. We overcome this using the self-attention mechanism. We first describe the proposed MT-DTI model architecture (Figure 4.1) with input and output representation. We then elaborate on each of the three main building blocks of our MT-DTI model, the character-embedded Transformer layers (Molecule Transformers, Figure 4.2a, Section 4.4.2), the character-

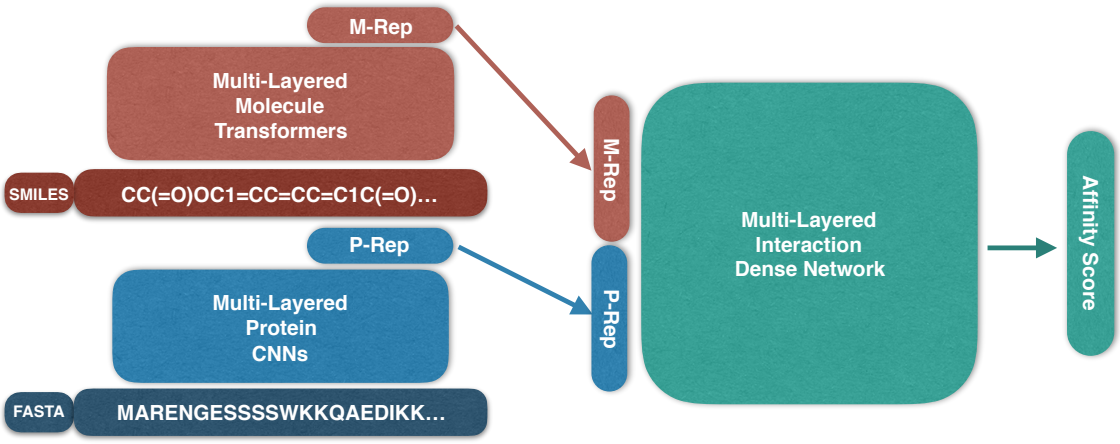


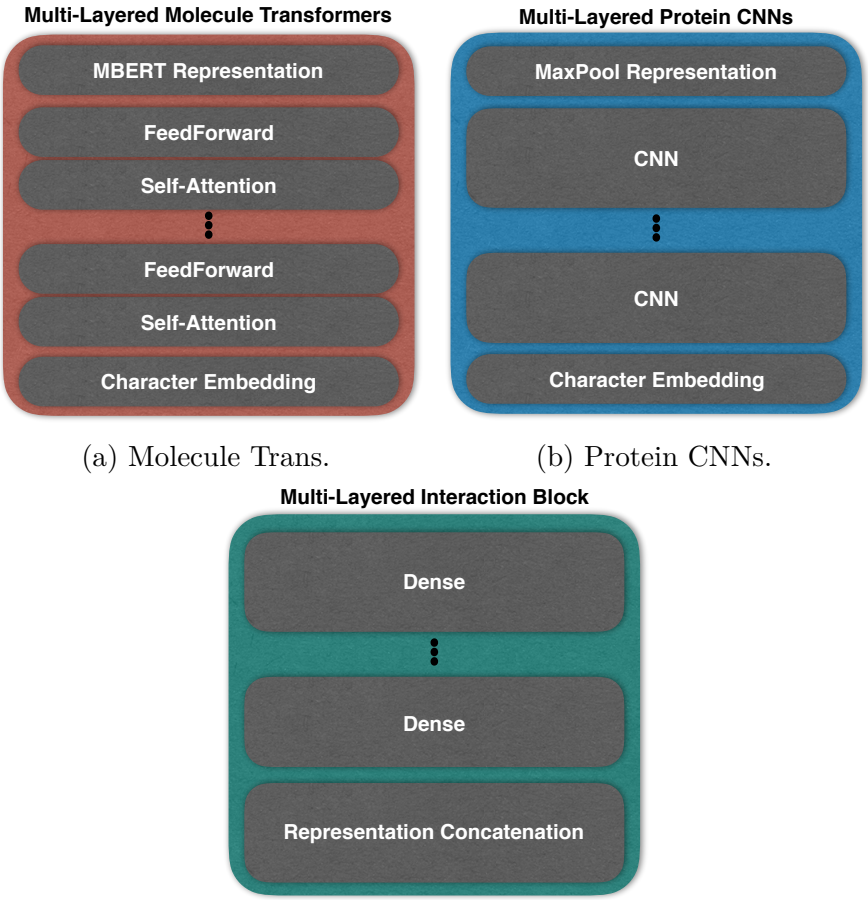
Figure 4.1: The Proposed DTI Model Architecture. Inputs are molecule (SMILES) and protein (FASTA) and the regression output is the affinity score between these two inputs.

embedded Protein CNN layers (Protein CNNs, Figure 4.2b, Section 4.4.3), and the dense layers to model interactions between a drug and a protein (Interaction Denses, Figure 4.2c, Section 4.4.4). Then, we explain the process for pre-training the molecule transformers (MT) (Section 4.4.2).

4.4.1 Model Architecture

The MT-DTI model takes two inputs: a molecule represented by the SMILES [165] sequence and a protein represented by the FASTA [94] sequence. A molecule represented using the SMILES sequence is comprised of characters representing atoms or structure indicators. Mathematically, a molecule is represented as $I_M = \{m_1, m_2, \ldots, m_{L_M}\}$, where m_i could be either an atom or a structure indicator, and L_M is the sequence length, which varies depending on a molecule. This molecule sequence is fed into the Molecule Transformers (Section 4.4.2), to produce a molecule encoding, $M_{enc} \in \mathbb{R}^{E_M}$. Another type of input, a protein with FASTA sequence, also consists of characters of various amino acids. A formal protein representation is $I_P = \{p_1, p_2, \ldots, p_{L_P}\}$, where p_j is one of the amino acids, and L_P is the sequence length, which varies depending on a protein. This protein sequence is the input of the Protein CNNs (Section 4.4.3) and

generates a protein encoding, $P_{enc} \in \mathbb{R}^{E_P}$. Note that the encoding vector dimension E_M and E_P are model parameters. Both of the encodings, M_{enc} and P_{enc} are together fed into the multi-layered feed-forward network, Interaction Denses (Section 4.4.4), followed by the last regression layer, which predicts the binding affinity scores.



(a) Molecule Trans. (b) Protein CNNs.

(c) Interaction Denses.

Figure 4.2: Three parts of the proposed model.

4.4.2 Molecule Transformers

Molecule Transformers (Figure 4.2a) are multi-layered bidirectional Transformer encoders based on the original Transformer model [160]. The Transformer can model a sequence by itself without using a recurrent neural network (RNN) or CNN. Unlike these previous sequence processing layers (RNN or CNN), Transformer can effectively

encode the relationship among long-distance tokens (atoms) in a sequence. This powerful context modeling enables many Transformer-based NLP models to outperform previous methods in many benchmarks [160, 41]. Molecule Transformers is a modification of the existing Transformer, BERT [41], to better represent a molecule by changing the cost function. Before plugging it into the proposed model (Figure 4.1), we pre-train it using the modified masked language model task, which was introduced in the BERT model [41]. Each Transformer block consists of a self-attention layer and feedforward layer, and it takes embedding vectors as an input. Therefore the first Transformer block needs to convert an input sequence into the form of vectors using the input embedding.

Input Embedding

The input to the Molecule Transformers is the sum of the token embeddings and the position embeddings. The token embeddings are similar to the word embeddings [108], in that each token, m_i is represented by a molecule token embedding (MTE) vector, e_i. These vectors are stored in a trainable weights MTE $\in \mathbb{R}^{V_M \times D_M}$, where V_M is the size of the SMILES vocabulary and D_M is the molecule embedding size. A MTE vector itself is not sufficient to represent a molecule sequence with a self-attention network, because a self-attention doesn't consider the sequence order when calculating the attentions, unlike other attention mechanisms. Therefore, we add a trainable positional embedding (PE)[2], $p_i \in \mathbb{R}^{L_M^{max} \times D_M}$, to e_i that makes the final input representation, x_i where L_M^{max} is the maximum length of a molecule sequence, which is set to 100 in this study. This process is illustrated in Figure 4.3.

We add five special tokens to the SMILES vocabulary to make a raw molecule sequence compatible with our model. [PAD] is for dummy padding to ensure the sequence has a fixed length. [REP] is a representation token that is used when fine-

[2]Please refer to [41] for more details.

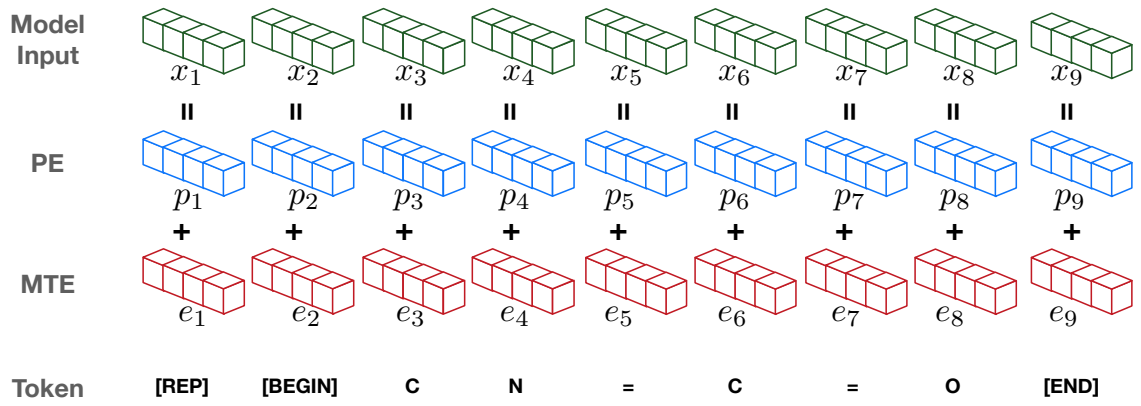


Figure 4.3: An example of molecule token embedding (MTE) and positional embedding (PE) to make the model input x_i for a given molecule sequence of methyl isocyanate (CN=C=O).

tuning the Transformer in the proposed MT-DTI model. [BEGIN]/[END] indicate the beginning or end of the sequence. These tokens are useful for the model when dealing with a sequence longer than L_M^{max}. When it is truncated on both sides, the absence of [BEGIN]/[END] tokens serve as an effective indicator of a truncation. Methyl isocyanate (CN=C=O), for example, can be represented with 9 tokens;

[REP] [BEGIN] C N = C = O [END]

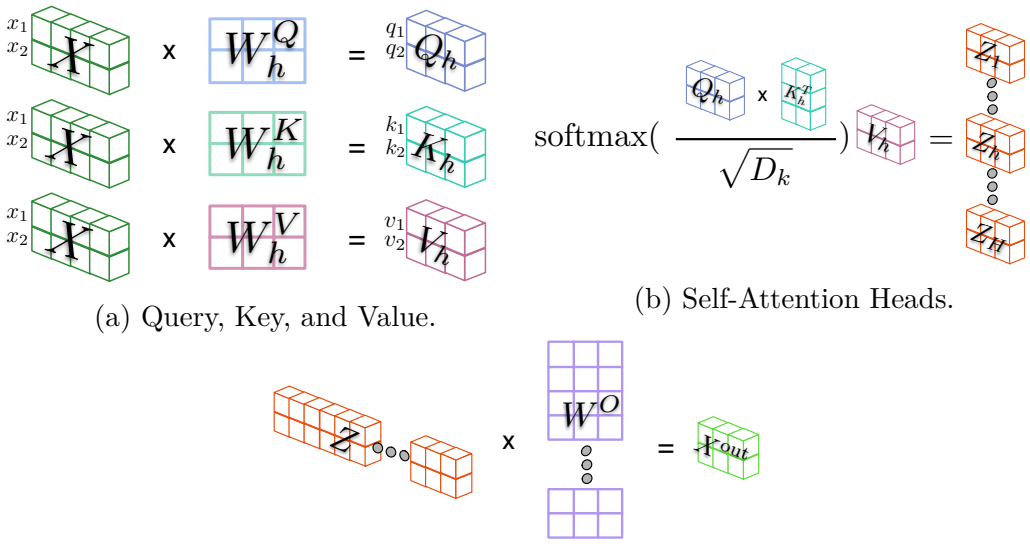
Each token is transformed into a corresponding vector by referencing MTE and PE.

Self-Attention Layer

These transformed input vectors, x_i, are now compatible with an input to a self-attention layer. Each self-attention layer is controlled by a query vector (q_i), key vector (k_i), and value vector (v_i), where $i \in \{0, 1, \ldots, L_M^{max}\}$, all of which are different projections of the input, X ($x_i \in \mathbb{R}^{L_M^{max} \times D_M}$), using trainable weights, $W^Q \in \mathbb{R}^{D_M \times D_q}$, $W^K \in \mathbb{R}^{D_M \times D_k}$, and $W^V \in \mathbb{R}^{D_M \times D_v}$, shown correspondingly in Figure 4.4a. Then, the attention weights are computed as:

$$Z = \text{Attention}(Q, K, V) \tag{4.1}$$
$$= \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right) V \in \mathbb{R}^{L_M^{max} \times D_v}$$

D_k is the dimension of the key (one of the Z's in Figure 4.4b). Thus, the learned relationship between the atoms can span the entire sequence via the self-attention weights.



(a) Query, Key, and Value.

(b) Self-Attention Heads.

(c) Projected Output.

Figure 4.4: Put your caption here

Feed-Forward Layer

Similar to multiple filters in convolutional networks, a Transformer can have multiple attention weights, called multi-head attention. If one model has H-head attention, then it will have $Z_h = \text{Attention}(XW_h^Q, XW_h^K, XW_h^V)$, where $h \in \{1, 2, \ldots, H\}$. These H number of attention matrices, Z_h, are then concatenated (shown on the left of Figure 4.4c) and projected using $W^O \in \mathbb{R}^{H \cdot D_v \times D_M}$ (shown on the middle of Figure 4.4c) to form a final output of a Transformer, $X^{out} \in \mathbb{R}^{L_M^{max} \times D_M}$ (shown on the right of Figure 4.4c).

pre-training

We adopt one of the pre-training tasks of BERT [41], the Masked Language Model. Since the structure of molecule sequences are shown to be very similar to the structure of natural language sentences [76] and there are abundant training examples, we hypothesize that predicting masked tokens is an effective way of learning a chemical structure. We adopt a special token, [MASK], for this task. It replaces a small portion of tokens so that the task of the pre-training model is to predict the original tokens. We choose 15% of SMILES tokens at random for each molecule sequence, and replace the chosen token with one of the special tokens, [MASK] with the probability of 0.8. For the other 20% of the time, we replace the chosen token with a random SMILES token[3] or preserve the chosen token, with an equal probability, respectively. The target label of the task is the chosen token with the index. For example, one possible prediction task for Methyl isocyanate (CN=C=O) is

$$\text{input} : \text{[REP] [BEGIN] C N = [MASK] = O [END]}$$

$$\text{label} : (\text{C}, 5)$$

Fine-tuning

The weights of the pre-trained Transformers (Section 4.4.2) are used to initialize the Molecule Transformers in the proposed MT-DTI model (Figure 4.1). The output of the Transformers is a set of vectors, where the size is equivalent to the number of tokens. To obtain a molecule representation with a fixed length vector, we utilize the vector of the special token, [REP] in the final layer. This vector conveys the comprehensive bidirectional encoding information for a given molecule sequence, denoted as $M^{rep} \in \mathbb{R}^{D_M}$.

[3]Since the [MASK] token does not exist when testing, we need to occasionally feed irrelevant tokens when training.

4.4.3 Protein CNNs

Another type of input to the proposed MT-DTI model is a protein sequence. We modified the protein feature extraction module introduced by [114] by adding an embedding layer for the input.[4] It consists of multi-layer CNNs with an embedding layer to make a sparse protein sequence continuous, and a pooling layer to represent a protein as a fixed size vector. For a given protein sequence, I_p, each protein token, p_j is converted to a protein embedding vector by referencing trainable weights, PTE $\in \mathbb{R}^{V_P \times D_P}$, where V_P is the size of the FASTA vocabulary and D_P is the protein embedding size. Let $P \in \mathbb{R}^{L_P^{max} \times D_P}$ be a matrix representing the input protein, where L_P^{max} is the maximum length of a protein sequence, which is set to 1000 in this study. This protein matrix P is fed into the first convolutional layer and convolved by the weights $c_1 \in \mathbb{R}^{s_1 \times D_P}$, where s_1 is the length of the filter. This operation is repeated m_1 times with the same filter length. Then this first convolution layer produces a vector $PC_1 \in \mathbb{R}^{L_P^{max}-s_1+1}$, where elements in PC_1 convey the s_1-gram features across the sequence. Multiple convolutional layers can be stacked on top of the previous output of the convolutional layer. After v number of convolution layers, the final vector, $PC_1 \in \mathbb{R}^{(L_P^{max}-s_1-s_2 \cdots -s_v+v) \times m_v}$, is fed into the max pooling layer. This max pooling layer selects the most salient features from the vectors produced by the filters from the last layer. Then, the output of this max pooling layer is a vector $P^{rep} \in \mathbb{R}^{D_P}$ ($m_v = D_P$).

4.4.4 Interaction Denses

A molecule representation ($M^{rep} \in \mathbb{R}^{D_M}$, Section 4.4.2) and a protein representation ($P^{rep} \in \mathbb{R}^{D_P}$, Section 4.4.3) are concatenated to create the input of Interaction Denses, $MP^{rep} \in \mathbb{R}^{D_M+D_P}$. Interaction Denses approximates the affinity score through a multi-layered feed-forward network with dropout regularization. The final

[4] Adding an embedding layer slightly improves the accuracy of the DTI model.

Dataset	# of Compounds	# of Proteins	# of Interactions	TRN	DEV	TST
DAVIS	68	442	30,056	20,037	5,009	5,010
KIBA	2,111	229	118,254	78,836	19,709	19,709

Table 4.1: Statistics of the Davis and Kiba datasets. TRN/DEV/TST: training, development, evaluation sets.

layer is a regression layer associated with the regression task for the proposed MT-DTI model. The weights of the network are then optimized according to the mean square error between the network output ($\hat{y}$) and actual affinity values (y).

4.5 Experiments

4.5.1 Datasets

Drug-Target Interaction

The proposed MT-DTI model is evaluated on two benchmarks, Kiba [152] and Davis [35], because they have been used for evaluation in previous drug-target interaction studies [116, 66, 114]. Davis is a dataset comprised of large-scale biochemical selectivity assays for clinically relevant kinase inhibitors with their respective dissociation constant (K_d) values. The original K_d values are transformed into log space, pK_d, for numerical stability, as suggested by [66] as follows:

$$pK_d = -\log_{10}(\frac{k_d}{1e9})$$

While Davis measures a bioactivity from one source of score, K_d, Kiba combines heterogeneous scores, K_i, K_d and IC_{50} by optimizing consistency among them. SimBoost [66] filtered out proteins and compounds with less than 10 interactions for computational efficiency, and we follow this procedure for a fair comparison. The number of compounds, proteins and interactions of the two datasets are summarized in Table 4.1. To facilitate comparison and reproducibility, we followed the same 5-fold

cross validation sets with a held-out test set which is publicly available[5].

Pre-training Dataset

We downloaded the chemical compound information from the PubChem database [55][6]. Only canonical SMILES information were used to maintain consistency of representation. A total of 97,092,853 molecules are available in the canonical SMILES format.

Drugbank Database

The DrugBank database comprises a bioinformatics and cheminformatics resource that provides known drug-target interaction pairs. To prove the effectiveness of drug candidates generated by our model, we designed a case study (Section 4.6.1) using this database. We extracted 1,794 drugs from the database, excluding any compounds that were used when training our model. These selected compounds were the input to the trained model (by Kiba dataset) along with a specific protein to generate corresponding Kiba scores. The scores were used to find the best candidate drugs targeting that protein.

4.5.2 Training Details

Molecule Transformer is first trained with the collected compounds from the PubChem database (Section 4.5.1), and then the trained Transformer is plugged into the MT-DTI model for fine-tuning.

Pre-training

We use 97 million molecules for pre-training. Before feeding it to the Molecule Transformer, we tokenize each molecule at the character level. If the length of the molecules

[5] `https://github.com/hkmztrk/DeepDTA/`

[6] `ftp://ftp.ncbi.nlm.nih.gov/pubchem/Compound/CURRENT-Full/SDF/`

is more than 100, we truncate its head and tail together to have a fixed size of 100. We choose the middle part of the longer sequence so that the model can easily distinguish truncated sequences by simply looking at the existence of [BEGIN] and [END] tokens. The network structure of the Molecule Transformer is as follows. The number of layers is 8, the number of heads is 8, the hidden vector size is 128, the intermediate vector size is 512, the drop-out rate is 0.1, and the activation is Gelu [67]. These parameters are picked from preliminary experiments and the hyperparameters used in the NLP model, BERT [41]. We hypothesized that finding a chemical structure might be roughly 2-4 times easier task than finding a language model, because the size of the SMILES vocabulary is smaller than natural languages (70 vs 30k). Although the SMILES vocabulary is 400 times simpler, the number of tokens in the PubChem molecule datasets is about 2.4 times more than what BERT used to pre-train (8B vs 3.3B). This indicated that the molecules might have more complexity than expected when only considering the size of the vocabulary. Therefore we used parameters that were 2-4 times smaller than BERT. We note that there may be other parameter sets that can yield even better performance. We use the batch size of 512 and the maximum token size of 100, which enables it to process 50K tokens in one batch. Considering the average length of the compound sequence is around 80, there are approximately 8 billion tokens in the training corpus. We pre-train Molecule Transformer for 6.4M steps, which is equivalent to 40 epochs (8B/50K*40=6.4M). With an 8-core TPU machine, the pre-training took about 58 hours. The final accuracy of the Masked LM task was about 0.9727, which is comparable to the 0.9855 achieved by BERT on natural language.

Fine-tuning

The specifications of the Molecule Transformer in the MT-DTI model are the same as the one used when pre-training (Section 4.5.2). The Protein CNNs (Section 4.4.3)

consists of one embedding layer, three CNN layers, and one max pooling layer. It uses 128-dimensional vectors for the embedding layer. For CNN blocks, we denote the filter size as K and the number of the filter as L. The final model parameter settings of CNNs are $K_1, K_2, K_3 = 12(\text{Kiba}), 8(\text{Davis})$ and $L_1 = 32, L_2 = 64, L_3 = 96$. The max pooling layer selects the best token representations from the last CNN layer, which makes the feature length as 96. Interaction Dense (Section 4.4.4) is comprised of three feed-forward layers and one regression layer. The layer sizes, when training Kiba, are 1024, 1024, 512 in order of the feature input to the regression layer and the learning rate, γ, is 0.0001. We reduce the network complexity when training Davis due to the small number of training samples. We use two feed-forward layers of sizes 1024 and 512. The learning rate is adjusted to 0.001. The entire network uses the same dropout rate of 0.1. All the hyper-parameters are tuned based on the lowest mean square error of the development sets for each fold, and the final score is evaluated on the held-out test set with the model at 1000 epochs.

4.5.3 Evaluation Metrics

We use four metrics to evaluate the proposed model: mean squared Error (MSE), concordance index (CI) [59], r_m^2, and area under the precision-recall curve (AUPR). MSE is a typical loss in the optimizer. CI is the probability that the predicted scores of two randomly chosen drug-target pairs, y_i and y_j, are in the correct order:

$$\text{CI} = \frac{1}{N} \sum_{y_i > y_j} h(\hat{y}_i > \hat{y}_j),$$

where N is a normalization constant (the number of data pairs) and $h(\cdot)$ is a step function [114]:

$$h(x) = \begin{cases} 1, & x > 0 \\ 0.5 & x = 0 \\ 0, & \text{else} \end{cases}$$

The r_m^2 [125, 135] index is a metric for quantitative structure-activity relationship models (QSAR models). Mathematically,

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

where r^2 and r_0^2 are the squared correlation coefficients with and without intercept, respectively. An acceptable model should produce an r_m^2 value greater than 0.5. Since AUPR is a metric for binary classification, we transform the regression scores to binary labels using known threshold values [66, 152]. For Davis, pairs with $pK_d \geq 7$ are marked as binding (1), others as no binding (0), and for Kiba, pairs with KIBA score ≥ 12.1 are marked as binding (1), others as no binding (0).

4.5.4 Baselines

For the baseline methods, two similarity-based models and one deep learning-based model, the current SOTA, are tested. One of the similarity-based models is KronRLS [116], whose goal is to minimize a typical squared error loss function with a special regularization term. The regularization term is given as a norm of the prediction model, which is associated with a symmetric similarity measure. Another similarity-based model is Simboost [66], which is based on a gradient boosting machine. Simboost utilizes many kinds of engineered features, such as network metrics, neighbor statistics, PageRank [115] scores, and latent vectors from matrix factorization. The last one is a deep learning model, which is the SOTA method in predicting

Datasets	Method	CI (std)	MSE	r_m^2 (std)	AUPR (std)
Kiba	KronRLS	0.782 (0.001)	0.411	0.342 (0.001)	0.635 (0.004)
	SimBoost	0.836 (0.001)	0.222	0.629 (0.007)	0.760 (0.003)
	DeepDTA	0.863 (0.002)	0.194	0.673 (0.009)	0.788 (0.004)
	MT-DTI$^{w/oFT}$	0.844 (0.001)	0.220	0.584 (0.002)	0.789 (0.004)
	MT-DTI	**0.882(0.001)**	**0.152**	**0.738(0.006)**	**0.837(0.003)**
Davis	KronRLS	0.871 (0.001)	0.379	0.407 (0.005)	0.661 (0.010)
	SimBoost	0.872 (0.002)	0.282	0.644 (0.006)	0.709 (0.008)
	DeepDTA	0.878 (0.004)	0.261	0.630 (0.017)	0.714 (0.010)
	MT-DTI$^{w/oFT}$	0.875 (0.001)	0.268	0.633 (0.013)	0.700 (0.011)
	MT-DTI	**0.887(0.003)**	**0.245**	**0.665(0.014)**	**0.730(0.014)**

Table 4.2: Test set results of the proposed MT-DTI model, MT-DTI model without fine-tuning (denoted as MT-DTI$^{w/oFT}$), and other existing approaches.

drug-target interactions, called DeepDTA [114]. It is an end-to-end model that takes a pair of sequences, (molecule, protein), and directly predicts affinity scores from the model. Features are automatically captured through back propagation of the multi-layered convolutional neural networks.

4.5.5 Results

The comparisons of our proposed MT-DTI model to the previous approaches are shown in Table 4.2. Reported scores are measured on the held-out test set using five models trained with the five different training sets. The best model parameters are selected based on the development set scores. MT-DTI outperforms all the other methods in all of the four metrics. The performance improvement is more noticeable when when there are many training data where the improvements of Kiba are 0.019, 0.042, 0.065, and 0.04 compared with Davis's improvements of 0.009, 0.016, 0.035, and 0.016, for CI, MSE, r_m^2, and AUPR, respectively. Furthermore, our model tends to be more stable with a larger training set, with the lowest standard deviation for CI and AUPR. Another interesting point is that our method without fine-tuning (MT-DTI$^{w/oFT}$ in Table 4.2) produced competitive results. It outperforms the similarity based metrics and performs better than Deep-DTA for some metrics. This suggests

that the molecule representation using pre-training learns some useful chemical structure that can be exploited by the interaction denses model.

4.6 Case Studies

4.6.1 Anticancer Drug Discovery

We performed a case study using actual FDA-approved drugs targeting a specific protein, epidermal growth factor receptor (EGFR). This protein is chosen because this is one of the famous genes related to many cancer types. We calculated the interaction scores between EGFR and the 1,794 selected molecules based on the DrugBank database (see Section 4.5.1 for the details). These scores are sorted in descending order and summarized in Table 4.3.

EGFR is a transmembrane protein that is activated by binding of ligands such as epidermal growth factor (EGF) and transforming growth factor alpha (TGFa) [68]. Mutations in the coding regions of the EGFR gene are associated with many cancers, including lung adenocarcinoma [145]. Several tyrosine kinase inhibitors (TKIs) have been developed for the EGFR protein, including gefitinib, erlotinib, and afatinib. More recently, Osimertinib was developed as a third generation TKI targeting the T790M mutation in the exon of the EGFR gene [149]. Since the direct binding of these drugs to EGFR protein is well known, we tested whether our proposed model can identify known drugs for the EGFR protein.

Biological Insights

The result indicated that our model successfully identified known EGFR targeted drugs as well as novel chemical compounds that were not reported for association with the EGFR protein. For example, the first and second generation TKIs, such as Erlotinib and Gefitinib, and Afatinib, respectively, were predicted to exhibit high

Ranking	Compound ID	Compound Name	KIBA Score
1	208908	**Lapatinib***	14.002403
2	11557040	**Lapatinib Ditosylate***	13.811217
3	10184653	**Afatinib***	13.404812
4	16147	Triamcinolone Acetonide Sodium Phosphate	13.147043
5	5485201	Naltrexone Hydrochloride	13.114577
6	123631	**Gefitinib***	13.111686
7	60699	Topotecan Hydrochloride	13.108758
8	5360515	Naltrexone	13.065864
9	441351	Rocuronium Bromide	13.032806
10	6918543	Almitrine Mesylate	13.016999
11	176870	**Erlotinib***	12.885199
12	23422	Tubocurarine Chloride Pentahydrate	12.87076
13	6000	Tubocurarine	12.809549
14	11954379	**Erlotinib Variant***	12.782704
15	11954378	**Erlotinib Hydrochloride***	12.768639
16	3389	Prolixin Enanthate	12.737285
17	23724988	Oxycodone Hydrochloride Trihydrate	12.709352
18	14676	Methdilazine Hydrochloride	12.662965
19	5281065	Ibutilide Fumarate	12.650397
20	9869929	Avanafil	12.635439
21	60700	Topotecan	12.618897
22	5360733	Nalbuphine Hydrochloride	12.610958
23	5282487	Paroxetine Hydrochloride Hemihydrate	12.608804
24	66259	Oxymetazoline Hydrochloride	12.557486
25	5311066	Desonide	12.538858
26	2247	Astemizole	12.536284
27	11954293	Asenapine	12.534941
28	11304743	Riociguat	12.527533
29	82153	Flunisolide	12.527164
30	71496458	**Osimertinib***	12.507524

Table 4.3: Compound ranking based on the predicted Kiba scores when the target is EGFR protein. All compounds are from Drugbank database excluded any compounds in Kiba dataset. [**Compound Name**]* represents a known EGFR targetting drug.

affinity to the EGFR protein (Table 3). Lapatinib [104], which inhibits the tyrosine kinase activity associated with two oncogenes, EGFR and HER2/neu (human EGFR type 2), was predicted to exhibit the highest affinity. Osimertinib was also identified. Interestingly, chemical compounds targeting opioid receptors (naltrexone hydrochloride, nalbuphine hydrochloride, and oxycodone hydrochloride trihydrate) for pain relief, antihistamines (methdilazine hydrochloride and astemizole), antipsychotic medication for schizophrenia (Prolixin Enanthate and Asenapine), and corticosteroids for skin problems (Triamcinolone acetonide sodium phosphate, Oxymetazoline hydrochloride, Desonide) were predicted to be associated with EGFR. Among these chemical compounds, Astemizole was suggested as a promising compound when treated with known drugs for lung cancer patients [47, 37]. Therefore, further investigations of these chemicals may provide a new therapeutic strategy for lung cancer patients.

4.6.2 Antiviral Drug Discovery

We performed another case study to suggest candidate molecules for the novel coronavirus found in Wuhan of China (COVID-2019). As the infection of COVID-2019 is rapidly spreading and there is a lack of effective treatment options for it, various strategies are being tested in many countries, including drug repurposing. In this case study, we used our MT-DTI model to identify commercially available drugs that could act on viral proteins of COVID-2019.

Target Proteins: We first narrow down potential target proteins that could inhibit the infection of the virus by referencing the previous coronavirus studies [88, 93, 20, 166, 170, 180, 91]. As a result, we extract the following six target sequences from the COVID-2019 whole genome sequence, from the National Center for Biotechnology Information (NCBI) database: 3C-like proteinase, RNA-dependent RNA polymerase, helicase, 3'-to-5' exonuclease, endoRNAse, and 2'-O-ribose methyltransferase.

Small molecules	K_d in nM
Atazanavir	94.94
Remdesivir	113.13
Efavirenz	199.17
Ritonavir	204.05
Dolutegravir	336.91
Asunaprevir	581.77
Ritonavir*	609.02
Simeprevir	826.24

Table 4.4: DTI prediction results of antiviral drugs available on markets against a COVID-2019 targeting 3C-like proteinase. Ritonavir is expressed in canonical and isomeric SMILES, and * indicates the isomeric SMILES of ritonavir.

Training Data: Since this case study is real world problem that requires a broader set of molecules and proteins, we pre-process and combine two bigdata sources in the literature: the Drug Target Common (DTC) database [153] and BindingDB [97] database. Three types of efficacy value, Ki, Kd, and IC_{50} are integrated by a consistence-score-based averaging algorithm (13) to make the Pearson correlation score over 0.9 in terms of Ki, Kd, and IC50. The MT-DTI model trained by these integrated dataset has the potential power to predict interactions between antiviral drugs and COVID-2019 proteins because it covers a wide variety of species and target proteins unlike DAVIS or KIBA that only include a small portion of human proteins.

Result: The COVID-2019 3C-like proteinase was predicted to bind with atazanavir (K_d 94.94 nM), followed by remdesivir, efavirenz, ritonavir, and other antiviral drugs that have a predicted affinity of $K_d > 100$ nM potency (Table 4.4).

No other protease inhibitor among antiviral drugs was found in the $K_d < 1{,}000$ nM range. Although there is no real-world evidence about whether these drugs will act as predicted against COVID-2019 yet, some case studies have been identified. For example, a docking study of lopinavir along with other HIV proteinase inhibitors of the CoV proteinase (PDBID 1UK3) suggests atazanavir and ritonavir, which are listed in the present prediction results, may inhibit the CoV proteinase in line with

Small molecules	K_d in nM
Grazoprevir	8.69
Ganciclovir	11.91
Remdesivir	20.17
Atazanavir	21.83
Daclatasvir	23.31
Acyclovir	26.66
Etravirine	33.09
Entecavir	52.83
Efavirenz	76.70
Asunaprevir	78.36
Abacavir	131.51
Dolutegravir	150.15
Lomibuvir	280.96
Penciclovir	312.93
Triflurdine	315.79
Danoprevir	405.66
Ritonavir	624.30
Saquinavir	704.86
Raltegravir	832.25
Lamivudine	999.92

Table 4.5: DTI prediction results of antiviral drugs available on markets against a COVID-2019 targeting RNA polymerase.

the inhibitory potency of lopinavir [36]. According to the prediction, viral proteinase-targeting drugs were predicted to act more favorably on the viral replication process than viral proteinase through the DTI model (Table 4.5-4.9). The results include antiviral drugs other than proteinase inhibitors, such as guanosine analogues (e.g., acyclovir, ganciclovir, and penciclovir), reverse transcriptase inhibitors, and integrase inhibitors.

Among the prediction results, atazanavir was predicted to have a potential binding affinity to bind to RNA-dependent RNA polymerase (K_d 21.83 nM), helicase (K_d 25.92 nM), 3'-to-5' exonuclease (K_d 82.36 nM), 2'-O-ribose methyltransferase (K_d of 390.67 nM), and endoRNAse (K_d 50.32 nM), which suggests that all subunits of the COVID-19 replication complex may be inhibited simultaneously by atazanavir (Table 4.5-4.9).

Small molecules	K_d in nM
Remdesivir	6.48
Simeprevir	23.34
Atazanavir	25.92
Grazoprevir	26.28
Asunaprevir	28.20
Telaprevir	40.75
Ritonavir	41.60
Lopinavir	78.49
Darunavir	90.38
Ganciclovir	108.21
Penciclovir	129.41
Etravirine	175.50
Raltegravir	299.81
Dolutegravir	333.32
Nelfinavir	365.96
Indinavir	401.78
Efavirenz	412.86
Entecavir	452.78
Ritonavir*	462.20
Boceprevir	510.35
Lomibuvir	543.41
Acyclovir	661.76

Table 4.6: DTI prediction results of antiviral drugs available on markets against a COVID-2019 targeting helicase.

Small molecules	K_d in nM
Simeprevir	13.40
Efavirenz	39.55
Remdesivir	45.20
Danoprevir	49.26
Ganciclovir	56.29
Penciclovir	71.76
Atazanavir	82.36
Entecavir	82.78
Daclatasvir	110.47
Grazoprevir	111.90
Asunaprevir	117.26
Ritonavir	182.51
Lomibuvir	182.65
Darunavir	195.73
Raltegravir	306.99
Dolutegravir	326.89
Lopinavir	959.76

Table 4.7: DTI) prediction results of antiviral drugs available on markets against COVID-2019 targeting 3'-to-5' exonuclease.

Small molecules	K_d in nM
Efavirenz	34.19
Atazanavir	50.32
Remdesivir	70.27
Ritonavir	124.36
Danoprevir	235.15
Grazoprevir	277.87
Dolutegravir	349.63
Lomibuvir	398.81
Lopinavir	472.08
Darunavir	562.40
Nelfinavir	576.82
Telaprevir	618.11
Abacavir	619.79
Raltegravir	727.37
Boceprevir	891.62

Table 4.8: DTI prediction results of antiviral drugs available on markets against a COVID-2019 targeting endoRNAse.

Small molecules	K_d in nM
Remdesivir	134.39
Atazanavir	390.67
Efavirenz	423.00
Boceprevir	433.93

Table 4.9: DTI prediction results of antiviral drugs available on markets against a COVID-2019 targeting 2'-O-ribose methyltransferase.

Also, ganciclovir was predicted to bind to three subunits of the replication complex of the COVID-19: RNA-dependent RNA polymerase (K_d 11.91 nM), 3'-to-5' exonuclease (K_d 56.29 nM), and RNA helicase (K_d 108.21 nM). Lopinavir and ritonavir, active materials of AbbVies Kaletra, both were predicted to have a potential affinity to COVID-19 helicase (Table 4.6) and are suggested as potential MERS therapeutics [38]. Recently, approximately $2 million worth of Kaletra doses were donated to China [70], and a previous clinical study of SARS by Chu et al. [26] may support this decision. Another anti-HIV drug, Prezcobix of Johnson & Johnson, which consists of darunavir and cobicistat, was to be sent to China [70], and darunavir is also predicted to have a K_d of 90.38 nM against COVID-19s helicase (Table 4.6). However, there was no current supporting literature found for darunavir to be used as a CoV therapeutic. Although remdesivir is not a FDA approved drug, its predicted potency to COVID-19 resulted as follows: against RNA-dependent RNA polymerase (K_d 20.17 nM), helicase (K_d 6.48 nM), 3'-to-5' exonuclease (K_d 45.20 nM), 2'-O-ribose methyltransferase (K_d of 134.39 nM), and endoRNAse (K_d 70.27 nM).

4.7 Discussion

This paper proposes a new molecule representation using the self-attention mechanism, which is pre-trained using publicly available big data of compounds. The trained parameters are transferred to our DTI model (MT-DTI) so that it can be fine-tuned using two DTI benchmark data. Experimental results show that our model outper-

forms all other existing methods with respect to four evaluation metrics. Moreover, the first case study of finding drug candidates targeting a cancer protein (EGFR) shows that our method successfully enlists all of the existing EGFR drugs in top-30 promising candidates. In addition, the antiviral drug discovery case study shows that the proposed method can produce useful results in a very short time.

This suggests our DTI model could potentially yield low-cost drugs and provide personalized medicines. Our model can be further improved as the proposed attention mechanism is also applied to represent proteins. However, we didn't explore this direction for two reasons. One reason is that the length of a protein sequence is ten times longer than a molecule sequence on average, which takes a considerable amount of time for computation. Another reason is the need for a protein dataset which contains enough sufficient information to pre-train the model. Unfortunately, such a dataset is not readily available.

4.8 Contribution

In this project, my contributions are as follows.

- Designed the study with Dr. Ho.
- (solely) Developed the algorithm
- Ran experiments with Park
- Analyzed the results with Dr. Kang and Dr. Ho

5

Molecule Generation

In this chapter we present another way of molecule discovery, molecule generation. The proposed method combines a molecule generation model and a molecule optimization model. We submitted this work to Knowledge Discovery in Databases conference 2020.

5.1 Introduction

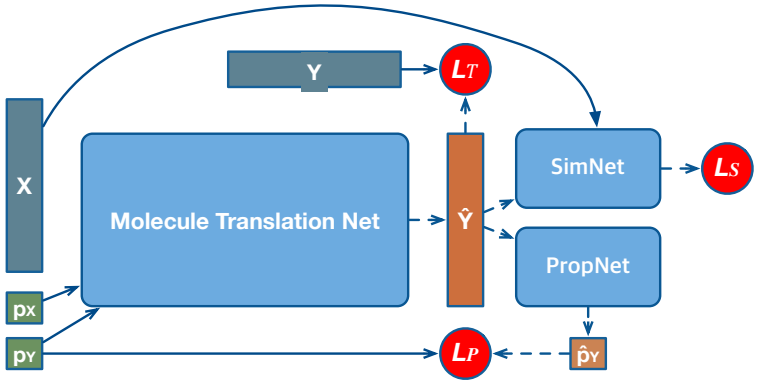
Drug discovery is an expensive process. According to DiMasi et al. [43], the estimated average cost to develop a new medicine and gain FDA approval is $1.4 billion. Among this amount, 40% of it is spent on the candidate compound generation step. In this step, around 5,000 to 10,000 molecules are generated as candidates but 99.9% of them will be eventually discarded and only 0.1% of them will be approved to the market. This inefficient nature of the candidate generation step serves as motivation to design an automated molecule search method. However, finding target molecules with the desired chemical properties is challenging because of two reasons. First, an efficient search is not possible because the search space is discrete to the input [85]. Second, the search space is too large that it reaches up to 10^{60} [124]. As such, this task is currently being tackled by chemistry and pharmaceutical experts and takes

years to design. Therefore, this study aims to accelerate the drug discovery process by proposing a deep-learning (DL) model that accomplishes this task effectively and quickly.

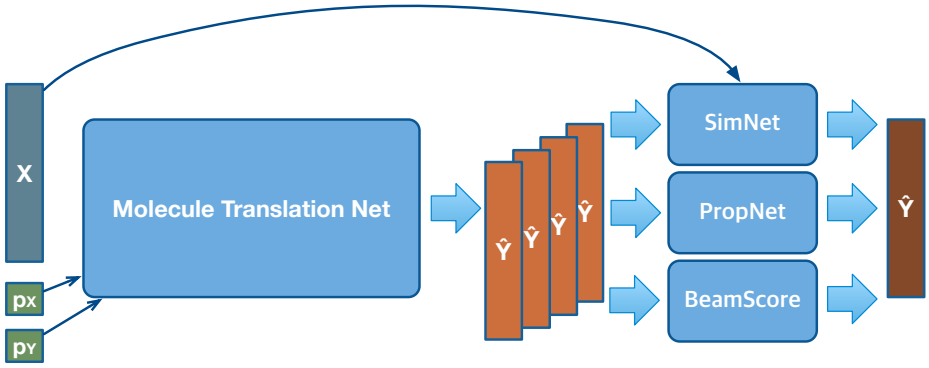
Recently, many methods of molecular design have been proposed [9, 138, 48, 57, 33, 89, 112, 63, 137, 175]. Among them, Matched Molecular Pair Analysis (MMPA) [62] and Variational Junction Tree Encoder-Decoder (VJTNN) [81] formulated molecular property optimization as a problem of molecular paraphrase. Just as a Natural Language Process (NLP) model produces paraphrased sentences, when a molecule comes in as an input to these models, another molecule with improved properties is generated by paraphrase. Although MMPA was the first to try this approach, it is not effective unless many rules are given to the model [81]. To mitigate this problem, Jin et al. [81] proposed VJTNN, an end-to-end molecule optimization model without the need for rules. By efficiently encoding and decoding a molecule with graphs and trees, it is the current state-of-the-art (SOTA) model for optimizing a single property (hereby referred to as a single-objective optimization task). However, it cannot optimize multiple properties at the same time (a multi-objective optimization task) because the model inherently optimizes only one property. As noted by Vogt et al. [162], Shanmugasundaram et al. [140], the actual drug discovery process frequently requires balancing of multiple compound properties.

With these motivations, we propose a new DL-based end-to-end model that can *optimize multiple properties in one model*. By extending the preceding problem formulation, we consider the molecular optimization task as a sequence-based controlled paraphrase (or translation) problem. The proposed model, controlled molecule generator (CMG), learns how to translate the input molecules given as sequences into new molecules as sequences that best reflect the properties of the molecules we want. Our model extends the Transformer model [160] that showed its effectiveness in machine translation. CMG encodes raw sequences through a deep network and decodes a new

molecule sequence by referencing that encoding and the desired properties. Since we represent the desired properties as a vector, this model inherently can consider multiple objectives simultaneously. Moreover, we present a novel loss function using pre-trained constraint networks to minimize generating invalid molecules. Lastly, we propose a novel beam search algorithm that incorporates these constraint networks into the beam search algorithm [105].



(a) The overall architecture at training.



(b) The overall architecture at prediction.

Figure 5.1: The proposed controlled molecule generator model.

We evaluate the proposed model using two tasks (single-objective optimization and multi-objective optimization) and two analysis studies (case study and ablation study)[1]. We compare our model with six existing approaches including the current SOTA, VJTNN. CMG outperforms all baseline models in both benchmarks. In addition, the our model shows the biggest diversity in the output molecule distribution.

[1]Code and data are available at `https://anonymous.url`

The case study demonstrates the practicality of our method through the target affinity optimization experiment using an actual experimental drug molecule. Lastly, the ablation study not only shows the effectiveness of each sub-part, but demonstrates the superiority of the proposed model itself without the sub-parts.

Contribution: The contributions of this paper are summarized as below;

- We formulate the multi-objective molecule optimization task as a sequence-based controlled molecule translation problem.
- We propose a new self-attention based molecule translation model that can reflect the multiple desired properties.
- We introduce new loss functions to incorporate the pre-trained constraint networks.
- We propose a novel beam search algorithm using the pre-trained constraint networks.
- We present how to curate appropriate training data to train the proposed model.

5.2 Related Work

Molecule property optimization: Molecule property optimization models can be divided into two types depending on the data representation: sequence representations and graph representations. One of the earlier approaches using sequence representations utilizes encoding rules [165], while the recent ones [57, 138, 89] are based on DL methods that learn to reconstruct the input molecule sequence. This is related to our work in terms of the input representation, but they offer subpar performance when compared to the SOTA models. Another group of research uses graph representations conveying structural information [32, 80, 136, 92, 34]. Among them, VJTNN [81] and MMPA [62, 44, 34] are closely related to our work because

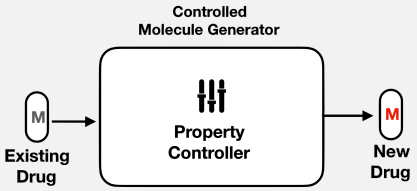
they formulate the molecule property optimization task as a molecule translation problem. From the model perspective, MMPA is a rule-based model and VJTNN is a supervised DL model. Although our approach is also based on a DL method, there is a big difference in practical use cases. A single VJTNN model is capable of optimizing a single property, while the proposed model can optimize multiple properties by using the controlled decoder. With these differences, we formulate the molecule property optimization task as a "controlled" molecule "sequence" translation problem. Other molecule generation methods include Junction Tree Variatinal Auto Encoder (JT-VAE) [79], Variational Sequence-to-Sequence (VSeq2Seq) [57, 5], Graph Convolutional Policy Network (GCPN) [178], and Molecule Deep Q-Networks (MolDQN) [184].

Natural Language Generation Model: Our model is inspired by the recent success in molecule representation using the self-attention technique [143]. By adopting the BERT [41] architecture to represent molecule sequences, their model becomes the SOTA in the drug-target interaction task. In terms of the model architecture, our work is related to Transformer [160] because we extend it to be applicable to the molecule optimization task. There is a controlled text generation model [72] in NLP domain. It is related to ours because they feed the desired text property as one of the inputs. However, all of these methods are designed for NLP tasks, therefore, they cannot be directly applied to molecule optimization tasks.

Transfer learning: DL-based transfer learning by pre-training has been applied to many fields such as computer vision [133, 54], NLP [71], speech recognition [75, 99], and health-care applications [141]. They are related to ours because we also pre-train the constrained networks and transfer the weights to the main model.

5.3 Problem Definition

Controlled Molecule Generation



- **Data**: Molecules in SMILES format
  - **X**: Molecule
  - **pX**: The property vector of the input molecule, X
  - **Y**: New molecule
  - **pY**: The property vector of the output molecule, Y
- **Task**: Produce a new molecule, Y, that best reflects pY by optimizing X

Given an input molecule X, its associated molecule property vector p_X, and the desired property vector p_Y, the goal is to generate a new molecule Y with the property p_Y with the similarity of $(X, Y) \geq \delta$. Note that δ is a similarity threshold and the similarity measure is Tanimoto molecular similarity over Morgan fingerprints [131]. Formally, for two Morgan fingerprints, F_X and F_Y, where both of them are binary vectors, the Tanimoto molecular similarity is defined as,

$$sim(F_X, F_Y) = \frac{|F_X \cap F_Y|}{|F_X \cup F_Y|} \tag{5.1}$$

5.4 Proposed Model: Controlled Molecule Generator

In this section, we introduce the proposed model, controlled molecule generator (CMG), for generating new molecules with user-specified desired properties and sim-

ilarity to an input molecule.

5.4.1 Model Overview

Our model extends the Transformer [160] to a molecular sequence by incorporating molecule properties and additional regularization networks. Inspired by the previous success in applying the self-attention mechanism to represent a molecule sequence [143], we treat each molecule just like a natural language sequence. As noted by [76], the context and structure information of atoms is important when understanding the properties of molecules, just as we use context and structure information when understanding natural language.

However, this NLP technique cannot be directly applied, because the structure of the molecular sequence differs from natural languages, where the hierarchy is a letter-word-sentence. Not only that, there is no training data available that is collected for the molecule translation task, while there are ample datasets in the NLP domain. To fill these gaps, we propose the controlled molecule generation model (Figure 5.1) and present how we gather the training data for this network (Section 5.5.1). We optimize the proposed model using three loss functions as briefly shown in Figure 5.1(a). In addition, we propose two constraint networks (Section 5.4.4 and Figure 5.3), including property prediction network (Figure 5.3(a)), and similarity prediction network (Figure 5.3(b)) to train the model more accurately. Lastly, we also present how we modify the beam search algorithm [105] to best exploit the existing auxiliary networks (Section 5.4.5), as briefly shown in Figure 5.1(b).

5.4.2 Background

To efficiently present the idea of the proposed model, we briefly overview Transformer [160], the basic building block of the proposed model.

Input Embedding: For a given input sequence, $X = \{x_1, x_2, \cdots, x_i, \cdots, x_L\}, x_i \in$

$\mathbb{R}^V$, where L is the length of the sequence and V is the number of vocabulary, we transform each token into a continuous vector, which is the sum of a token embedding vector and the positional embedding vector. These token embeddings are similar to word embeddings [108] except they are randomly initialized, therefore, each token, x_i is transformed into $v_i \in \mathbb{R}^d$, where d is the token embedding size. The token embeddings themselves are not sufficient to represent a sequence with a self-attention network, because a self-attention doesn't consider the sequence order when calculating the attention, unlike other attention mechanisms. Therefore, we add a fixed positional embedding, $p_i \in \mathbb{R}^d$, to v_i that makes the final input representation, $e_i = v_i + p_i, e_i \in \mathbb{R}^d$.

Self-Attention Layer: These transformed vectors, e_i, are the inputs of the encoder, consisting of multiple stacks of a self-attention layer and a feed-forward network. Each self-attention layer possesses three dense networks; a query network ($f_{\theta_Q}, \theta_Q \in \mathbb{R}^{d\times h}$), key network ($f_{\theta_K}, \theta_K \in \mathbb{R}^{d\times h}$), and value network ($f_{\theta_V}, \theta_V \in \mathbb{R}^{d\times h}$), where h is the hidden dimension. With these three networks, each input vector, e_i is projected into three utility vectors, a query vector (q_i), key vector (k_i), and value vector (v_i). Now, the output of a self-attention layer is computed as:

$$\begin{aligned} S &= \text{Attention}(Q, K, V) \\ &= \text{softmax}(\frac{QK^T}{\sqrt{h}})V \in \mathbb{R}^{L\times h} \end{aligned} \tag{5.2}$$

This self-attention computation (Equation 5.2) can be repeated H number of times with the same input, forming the multi-head attention.

Feed-Forward Layer: The outputs of this multi-head attention are concatenated and projected using another dense network, called an intermediate dense layer, which parameter is represented as $\theta_O \in \mathbb{R}^{H\cdot h\times d}$. Then, it forms the final output of one encoder block, $o_i \in \mathbb{R}^d$.

Encoder: The Transformer encoder is multiple stacks of the two layers; the self-

attention layer and the feed-forward layer explained above. Note that the sequence length is preserved because the self-attention is applied to its own sequence, which preserves the input-output length. Therefore, the final output of the Transformer encoder is $z_i \in \mathbb{R}^d$, which is compatible to be an input to another encoder.

Decoder: The Transformer decoder includes not only the same sub-layers as the Transformer encoder but one additional layer, the cross attention layer. It is similar to the self-attention layer in that it's controlled by three vectors, (q_i, k_i, v_i). The difference is that its attention weights are calculated between the last unit's output of the Transformer encoder and each unit of the decoder, while the self-attention layer calculates its weights between the same layers.

Loss Function: The output of the last unit of the Transformer decoder is passed through the two dense layers, one for producing logits for all tokens and another for producing vocabulary probabilities for all tokens, $\hat{Y} = \{\hat{y_1}, \hat{y_2}, \cdots, \hat{y_j}, \cdots, \hat{y_M}\}$, where M is the number of the output tokens. Given an output sequence, $Y = \{y_1, y_2, \cdots, y_j, \cdots, y_N\}$, and the predictions, $\hat{Y}$, the loss function is a cross entropy that can be formally defined as:

$$\begin{aligned} &\mathcal{L}_T(\boldsymbol{\theta}_T; X, p_X, p_Y) \\ &\quad = -\frac{1}{N}\frac{1}{M}\sum_{n\in N}\sum_{j\in M}\sum_{v\in V} y_{v,j,n} \cdot \log(\hat{y}_{v,j,n}) \end{aligned} \tag{5.3}$$

$\boldsymbol{\theta}_T$ denotes all parameters of the Transformer and N represents the number of training samples.

5.4.3 Molecule Translation Network

In the molecule translation network, two major modifications are applied to the Transformer model [160]. The first change is applied to the molecule embedding. The input molecule is represented by the simplified molecular-input line-entry system (SMILES) sequence [165]. It is comprised of characters representing atoms or structure indica-

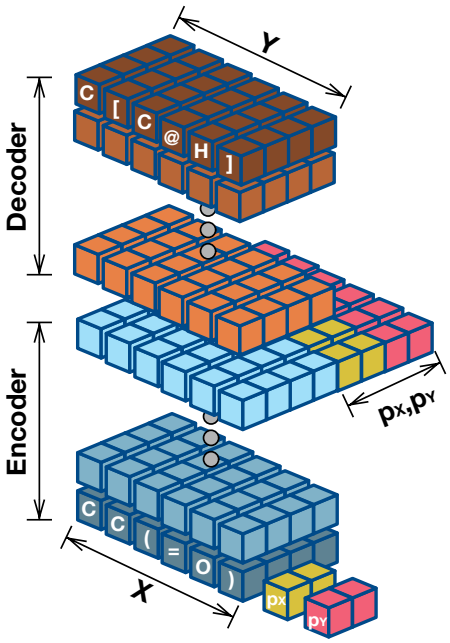


Figure 5.2: The molecule translation network.

tors. To mark the beginning and the end of a sequence, we add "[BEGIN]" token at the first position of the sequence and "[END]" token at the last. Mathematically, a molecule is represented as $X_L = \{x_1, x_2, \ldots, x_L\}$, where x_i could be either an atom or a structure indicator, and L is the sequence length, which varies depending on a molecule. Since the input sequence is a series of characters without any spaces, we tokenize this sequence into a list of a single token, then pass these tokens to the Transformer encoder. Another modification is that we add chemical property awareness to the hidden layer of the Transformer model. We enrich token vectors of the last encoder by concatenating property vectors to each of the token vectors as shown in Figure 5.2. Formally, let z_i be the token vectors in the last encoder. Then, the the new encoding vector becomes $z'_i = (z_i, p_X, p_Y) \in \mathbb{R}^{d+2k}$, where k represents the number of properties. Although it might be seen as a simple method, this empirically shows the best result among other types of configurations, such as property embeddings, disentangled encodings (property and non-property encodings), and concatenating property differential information instead of providing two raw vectors. The increased

vector size can be handled by adjusting the three weights of the projection network (θ_Q, θ_K, and θ_V) in the last encoding layer.

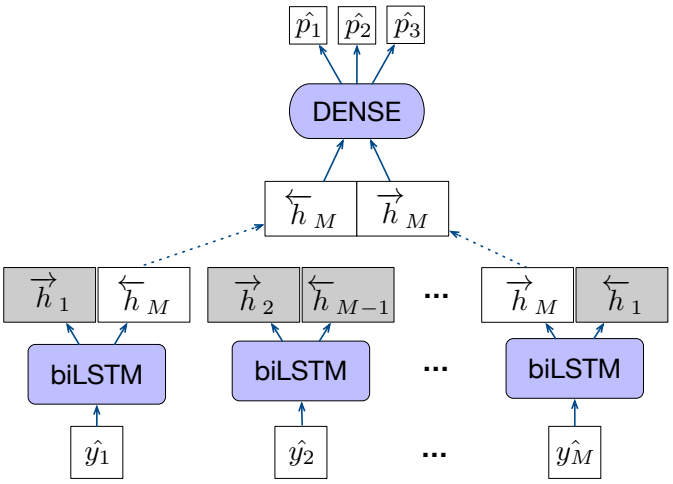
Curation of the Training Data: Since CMG is based on sequence translation, we need to appropriately curate the dataset. Similar to the previous work [81], we gather molecule pair (X, Y) so that similarity$(X, Y) \geq \delta$ from the available molecule list. Next, we calculate property scores p_X and p_Y using the third-party tool. The detailed information can be found in Section 5.5.

5.4.4 Constraint Networks

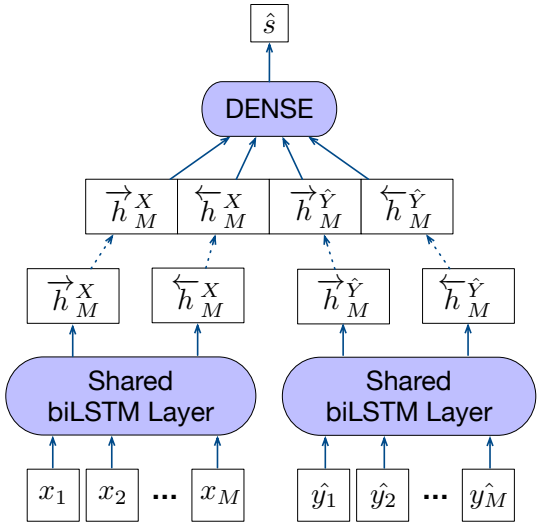
As described in Section 5.4.2, the cost function of the Transformer network (Equation 5.3) is the cross entropy between the target (y_i) and predicted molecule ($\hat{y}_i$). We hypothesize that this information is not enough to teach the generating model, because the error signals generated by that loss function can hardly capture the valuable information, such as if a predicted sequence pertains to the desired property or if it satisfies the similarity constraint. With this motivation, we add two constraint networks; the property prediction network (Section 5.4.4) and the similarity prediction network (Section 5.4.4).

Property Prediction Network

Since the model complexity of the Transformer is quadratic in the input length, additional networks should be as simple as possible and yet accurate. Therefore, we employ a single layer of Bidirectional Long Short Term Memory [69] (BiLSTM) network in the proposed constraint networks. As shown in Figure 5.3(a), the property prediction network (PropNet) takes the predicted molecule sequence ($\hat{y}_j$) as an input. A left-to-right LSTM layer encodes an input vector ($\hat{y}_j$) into a hidden vector, $\overrightarrow{h}_j \in \mathbb{R}^d$ by the following equations;



(a) Property Prediction Network. Gray indicates unused vectors.



(b) Similarity prediction network. The biLSTM layer is simplified because the details are described in (a). One biLSTM layer is being shared by two input sequences.

Figure 5.3: Two Constraint Networks.

$$i_j = \sigma(W_i Y[:j] + U_i h_{i-1} + b_i \tag{5.4}$$

$$f_j = \sigma(W_f Y[:j] + U_f h_{i-1} + b_f \tag{5.5}$$

$$C_j = i_i \odot \tanh(W_c Y[:j] + U_c h_{i-1} + b_c) + f_j \odot C_{j-1} \tag{5.6}$$

$$o_j = \sigma(W_o Y[:j] + U_o h_{i-1} + b_o \tag{5.7}$$

$$h_j = o_j \odot \tanh(C_j) \tag{5.8}$$

If we apply these same equations with different parameters on the opposite direction of the input, we get another hidden vector, $\overleftarrow{h}_j \in \mathbb{R}^d$. We concatenate the last two vectors from each direction, ($\overrightarrow{h}_M$ and $\overleftarrow{h}_M$), to create the property feature vector ($h_{prop} = (\overrightarrow{h}_M, \overleftarrow{h}_M) \in \mathbb{R}^{2d}$). This feature vector is fed into a dense network with two hidden layers. The output of the first dense layer has 100 neurons and the dimension of the next layer's output is the same as the property dimension. Since the property values are not strictly from 0 to 1, we don't apply any activation function to the last dense layer, while we apply the RELU activation on the first dense layer. Since we have the property prediction, $\hat{p_Y}$ and the desired property (p_Y) from the input, we can create another loss function, which will enrich error signals by adding property awareness in predicting a molecule. The loss function is formally written as,

$$\mathcal{L}_P(\boldsymbol{\theta}_T; X, p_X, p_Y) = \frac{1}{N} \sum_{n \in N} |p_{Y\,n} - \hat{p_Y}_n|^2 \tag{5.9}$$

Note that the parameters of PropNet are pre-trained using all molecules in the training set. Since all properties can be calculated using a third-party library, property annotations can be automated. Once pre-trained, the parameters are transferred to the CMG network. When training the main network, we freeze the weights of PropNet such that its role in the main network is consistent.

Implementation Details: We use 64-dimensional hidden vectors in the biLSTM layer, and 100 dimensions in the second last dense layer. We use Adam optimizer [84]

with default parameters set by Tensorflow Keras[2]. The batch size is 4,096 and the number of epochs is 1,000. The best model was selected by evaluating 20% of the data, the validation set of PropNet described above. As a result, we selected the model at the 715th epoch, the mean square error (MSE) on the test set is 0.08554. Considering the values of QED and DRD2 range from 0 to 1, and the values of penalized logp typically range from -10 to 10, this MSE is small enough to be used as a property estimation.

Similarity Prediction Network

Another constraint network is the similarity prediction network (SimNet). It takes two sequences as an input, one is a predicted molecule sequence ($\hat{y}_i$) and the other one is the input molecule sequence (x_i). Since one of the requirements of the model is to generate a molecule that is similar to input, we hypothesize that adding error signals to the loss function predicted by SimNet could be useful. We employ one layer of BiLSTM for SimNet, which is shared by two different inputs because two sequences need to have the same feature representation in order to be compared against each other. Applying the BiLSTM layer to a predicted molecule sequence will produce a feature vector of the predicted molecule, $h_{predicted} \in \mathbb{R}^{2d}$. Likewise, the feature vector of the input molecule is $h_{input} \in \mathbb{R}^{2d}$. We concatenate these two feature vectors as $(h_{predicted}, h_{input}) \in \mathbb{R}^{4d}$, so that the next two dense networks can capture the similarity between the two. The output of the first dense layer has 100 neurons with RELU activation, and the output of the next layer has one neuron with sigmoid activation because SimNet is designed for a binary classification task (similar or not). For the training dataset for SimNet, we annotate all training molecule pairs with either 1 (if similarity$\geq \delta$) or 0 (otherwise), by calculating the similarity using a third-party library. We let s_n be the binary label of the similarity between the two molecules

[2]`https://www.tensorflow.org/api_docs/python/tf/keras`

in n'th data, and $\hat{s}_n$ be the predicted output of SimNet for the same input data. With these labels and the predictions of SimNet, we can create the last loss function, formally written as,

$$\mathcal{L}_S(\boldsymbol{\theta}_T; X, p_X, p_Y) = \frac{1}{N} \sum_{n \in N} s_n \log \hat{s}_n + (1 - s_n) \log (1 - \hat{s}_n) \tag{5.10}$$

When training the whole model, we transfer the trained SimNet weights into the whole model and freeze the SimNet weights for the same reason as PropNet presented in Section 5.4.4. The SimNet network configuration is illustrated in Figure 5.3(b).

Implementation Details: We use 64-dimensional hidden vectors in the biLSTM layer, and 100 dimensions in the second last dense layer. We use the same optimizer as PropNet. The batch size is 4,096 and the number of epochs is 1,000. The best model was selected by evaluating 20% of the data, test set set of SimNet described above. As a result, we selected the model at the 755th epoch, which recorded the prediction accuracy of the test set as 97.59%.

The weights of these two constraint networks are transferred to the corresponding part in the main CMG model. These constraint networks in CMG are frozen when training CMG and predicting a new molecule using it.

CMG Loss Function

By combining Equations (5.3)-(5.10), we can obtain the CMG loss function:

$$\mathcal{L}_{CMG} = \mathcal{L}_T + \lambda_p \mathcal{L}_P + \lambda_s \mathcal{L}_S, \tag{5.11}$$

where λ_p and λ_s are weight parameters.

5.4.5 Modified Beam Search with Constraint Networks

When generating a sequence from CMG at testing, there is no gold output sequence that it can reference. Therefore we need to sequentially generate tokens until

Algorithm 2: Modified Beam Search

1: **Input:** Candidate molecules: $C_1, C_2, \cdots, C_b$,
Corresponding beam scores: $s_1, s_2, \cdots, s_b$
Input molecule: X
Desired property vector: p_Y
2: **for** $i = 1$ **to** b **do**
3: $\hat{p}_i \leftarrow \text{PropNet}(C_i)$
4: $p_d \leftarrow |p_Y - \hat{p}_i|$
5: $s_{pn} \leftarrow \text{reduce\_mean}(1 - p_d)$
6: $s_{sn} \leftarrow \text{SimNet}(X, C_i)$
7: $s_i \leftarrow s_i + (s_{pn} + s_{sn})$
8: **end for**
9: $\text{best\_index} \leftarrow \arg\max s_i$
10: **Output:** $C_{\text{best\_index}}$

we encounter the "[END]" token, like other sequence-based algorithms. For this process, a typical way is the beam search, where the model maintains top b number of best candidate sequences when predicting each token. When all candidate sequences are complete and ready, the model outputs the best candidate in terms of a beam score, a cumulative log-likelihood score for a corresponding candidate. We hypothesize that other non-best candidates might be better than the chosen one in terms of the purpose of a task. For example, there is a possibility that low ranked molecules could be closer to the desired properties than the top molecule selected by the beam search.

Unlike a typical Transformer model, CMG has PropNet and SimNet, that can be used to evaluate candidates before naively selecting one according to beam scores. Therefore, we propose a modified beam search algorithm using our constraint networks as summarized in Algorithm 2. It runs with b number of the completed candidates along with corresponding beam scores. For the property evaluation, we first get the predicted property of each candidate and get the absolute difference from the desired property (Line 3-4 in Algorithm 2). Since this difference is desired to be small, we calculate the property evaluation score (s_{pn}) by subtracting them from one

(Line 5 in Algorithm 2). The property could have multiple values, therefore, we take an average of all elements of this difference vector. For the similarity evaluation, we get the predicted similarity between the input X and each candidate C_i (Line 6 in Algorithm 2). Since we expect a candidate should be similar (label 1) to the input, we regard the predicted similarity as the raw score from SimNet. By adding these two predicted scores to the original beam scores, we obtain the modified beam scores (Line 7 in Algorithm 2). With this new score, we can select the best candidate (Line 9-10 in Algorithm 2).

5.4.6 Diversifying the Output

Unlike other variational models (VSeq2Seq and VJTNN), the proposed one encodes a fixed vector that is able to generate a single output for one input. In order to diversify the output for a fixed input, we re-parameterize the desired vector, (p_1, p_2, p_3), as random variable by adding a Gaussian noise with a user-specified variance.

$$\tilde{p}_k \sim N(p_k, \sigma_k) \tag{5.12}$$

For example, if the desired property vector is (p_1, p_2, p_3), we feed $(p_1+\alpha, p_2+\beta, p_3+\gamma)$, where α, β and γ are samples drawn from $N(0, \sigma_1)$, $N(0, \sigma_2)$, and $N(0, \sigma_3)$.

5.5 Experiments

We compare the proposed model with state-of-the-art molecule optimization methods in the following tasks.

Single Objective Optimization (SOO): This task is to optimize an input molecule to have a better property while preserving a certain level of similarity between the input molecule and the optimized one. Since developing a new drug usually starts with an existing molecule [8], this task serves as a good benchmark.

	ZINC	DRD2D	ZINC ∪ DRD2D
# of mol.	249,455	25,695	260,939

Table 5.1: Molecules used in this study. About 54% of DRD2D molecules also belong to ZINC, therefore, the total number of molecules for this study is 260,939.

Multi-Objective Optimization (MOO): This task reflects a more practical scenario in drug discovery, where modifying an existing drug involves optimizing multiple properties simultaneously, such as similarity, lipophilicity scores, drug likeness scores, and target affinity scores. Since improving one property might often result in sacrificing other properties, this task is harder than a single-objective optimization task.

To present multi-faceted aspects of the proposed model, we additionally perform the following case studies.

Case Study: To evaluate the effectiveness of the proposed model, we present the result of an actual drug optimization task with an existing molecule in an experimental phase.

Ablation Study: For ablation study, we report the validity of the constraint networks both in training and testing phases.

5.5.1 Datasets

Training Set for CMG: As described in Table 5.1, we use the ZINC dataset [150] and the DRD2 related molecule dataset (DRD2D) [112] for our experiments. This is the same set of molecules as what Jin et al. [81] used to evaluate their model (VJTNN). This set is chosen since Jin et al. [81] set up the benchmarks with this data, one of which (single-objective optimization) is used in our study. From these 260k molecules, we exclude molecules that appear in the development and the test set of VJTNN, resulting 257,565 molecules. With these molecules, we construct training datasets by selecting molecule pairs (X, Y) with the similarity is greater than or

equal to 0.4, following the same procedure in [81, 34]. The main difference from their curation processes is that we add all pairs of molecules even if any property is decreased. By doing this, we provide a more ample dataset to a deep model, so that it could be helpful in finding more useful patterns. As a result, the number of pairs in training data is significantly bigger than theirs. Among all possible pairs (257K $\times$ 257K = 67B), we select 10,827,615 pairs that satisfies similarity condition ($\geq$ 0.4). With the same similarity condition, Jin et al. [81] gathered less than 100K due to additional constraints of training sets, where the property values of each output molecule should be greater than the ones of the corresponding input. As previous related works [81, 89] did, we pre-calculate the following three chemical properties of all molecules in the training set:

- **Penalized logP (PlogP) [89]**: A measure of lipophilicity of a compound, specifically, the octanol/water partition coefficient (logP) penalized by the ring size and synthetic accessibility.
- **Drug likeness (QED)**: A measure of drug-likeness based on the quantitative estimate of drug-likeness proposed by Bickerton et al. [8]
- **Dopamine Receptor (DRD2)**: A measure of molecule activity against a biological target, the dopamine type 2 receptor.

Training Set for PropNet: Among 260,939 molecules, we excluded all molecules in the test sets of the two tasks; single-objective optimization, multi-objective optimization. The number of these remained molecules is 257,565. We construct the dataset for PropNet by arranging all molecules as inputs and the corresponding three properties as outputs. We randomly split this into the training and validation sets with a ratio of 8:2.

Training Set for SimNet: We use a subset of all 10,827,615 pairs in the CMG training set due to the simpler network configuration of SimNet. When sub-sampling

pairs, we tried to preserve the proportion of the similarity in the CMG dataset to best preserve the original data distribution. The reason behind this effort is that preserving the similarity distribution could possibly contribute to the SimNet accuracy although SimNet only uses binary labels. In addition, we try to preserve the similar/not-similar ratio to be about to the same. By sampling about 10% of data, we gathered 997,773 number of pairs and the ratio of the positive samples is 49.45%. We randomly split this into the training and validation set with a ratio of 8:2.

All molecules are represented in the SMILES format, and we parse them into a single character. Since the number of possible characters in the SMILES format is 71 and there are two special tokens, the size of the vocabulary is 73 when parsing a molecule data into a sequence of vocabulary ids.

5.5.2 Pre-Training of Constraint Networks

We pre-train the two constraint networks using the training sets described in Section 5.5.1. The best models were selected by evaluating 20% of each dataset, the validation sets. The weights of these two constraint networks are transferred to the corresponding part in the main CMG model. These constraint networks in CMG are frozen when training CMG and predicting a new molecule using it.

5.5.3 Single Objective Optimization

The first task on which we evaluate the proposed model is the single objective optimization task proposed by Jin et al. [79]. The goal is to generate a new molecule with an improved PlogP score under the similarity constraint ($\delta = 0.4$). We used the same development and test sets provided by Jin et al. [79]. The number of data samples in the development set and the test set are 200 and 800, correspondingly.

Baselines: We compare the proposed method with the following baselines; MMPA, JT-VAE, GCPN, VSeq2Seq, MolDQN, and VJTNN introduced in Section 5.2.

Since Jin et al. [81] ran and reported almost all of the baseline methods on the single property optimization task (PlogP improvement task) with the same test sets, we cite their experiment results. For MolDQN, which is published after VJTNN, we referenced the scores from MolDQN paper [184].

Following the same experimental setup with the baselines [81], we generate 20 molecules for a single input molecule and record one valid molecule with the maximum PlogP improvement.

Implementation Details: We use 4 layers and 8 heads of self-attention and feed-forward layers for both the encoder and the decoder. The hidden vector size is 128 and the dimension of the intermediate dense layer is 256. We set the maximum sequence length to be 150 because the max length used in the previous self-attention based molecule representation model [143] was 100 and the default buffer size of a typical Transformer model is 50% of its maximum sequence length. For the two constraint networks, we use the same configuration as pre-training models of them so that they are compatible with each other when transferring the weights. We use Adam optimizer [84] with the learning rate=2.0, $\beta_1 = 0.9$ and $\beta_2 = 0.997$. We train the proposed model (CMG) using 10M of the training set (Section 5.5.1) for 500 epochs with a batch size of 4,096. The dimension of the property vector is three, where the first one is PlogP, the second one is QED, and the last one is DRD2 values. We use the desired property vector of $\{X_{P\log P}, 0.0, 0.0\}$ with the sampled offset parameters of $\alpha = \{-1.0, -0.5, 0.0, 0.5, 1.0\}$, $\beta = \{0.1, 0.6\}$ and $\gamma = \{0.52, 0.8\}$.

Metrics: Since the task is to generate a molecule with an improved PlogP value, we measure an average of raw increments and its standard deviation among valid molecules with the similarity constraint met. Specifically, among 20 generated molecules for a given input, if none of them satisfies the similarity constraint ($\delta = 0.4$), then we score that sample as 0, otherwise, we choose one molecule with a maximum increment as VJTNN did [81]. After we repeat this for all 800 molecules in the test

Method	Improvement	Diversity
MMPA	3.29 ± 1.12	0.496
JT-VAE	1.03 ± 1.39	-
GCPN	2.49 ± 1.30	-
VSeq2Seq	3.37 ± 1.75	0.471
MolDQN	3.37 ± 1.62	-
VJTNN	3.55 ± 1.67	0.480
Proposed	**3.92 ± 1.88**	**0.545**

Table 5.2: Single objective optimization performance comparison on the penalized logP task. MolDQN results are from, and the scores of other baselines are from.

set, we report the average and standard deviation. In addition to the improvement score, we also measure the diversity defined by Jin et al. [81]. Although this diversity measure has been used by previous researches, it is limited in that it encourages the outputs to have low similarity around the threshold. However, if we need to generate diverse molecules around the similarity threshold, then this diversity measure can serve as the right metric.

The diversity measure is the average pairwise Tanimoto distance between the two molecules in each pair, where the distance is $dist(X, Y) = 1 - sim(X, Y)$. We measure the distances between an input molecule and validly generated molecules among 20 outputs. Then the final score is the average of them. With this metric, we can compare the diversity of learned output distributions.

Result: After we train the model using the training set described in Section 5.5.1, we generate new molecules by feeding inputs including desired chemical properties to the trained model. As discussed in Section 5.4.6, we add offsets to desired properties so that the output can be diversified. Since the number of generated samples for each input is set to 20, we use the desired property vector of $\{X_{P \log P}, 0.0, 0.0\}$ with a total of 20 combinations of (α,β,γ) that are sampled from the user-defined distributions. We select the best model using the development set, and the test set performance of that model is reported in Table 5.2. In the PlogP optimization task, the proposed model outperforms all baselines including the current SOTA, VJTNN, in terms of

both the average improvement and the diversity by a large margin. Considering the two recently proposed methods (MolDQN and VJTNN) are competing in 0.18 difference, the proposed one surpasses the current SOTA by 0.37 improvement. The same trend can be found in the diversity comparison. The proportion of valid output molecules generated by CMG is approximately 90%, while VJTNN records 100%. This indicates that once CMG produces valid molecules, then their improvements are big enough to compensate for loss from the invalid molecules (about 10% of all outputs).

We also experimented with other single objective optimizations, QED and DRD2. Unlike the PlogP case, where CMG outperforms the previous methods, these cases show that CMG failed to outperform others. The reason stems from the distribution of the training set. The proportions of pairs with the QED and DRD2 improvement in the training set are just 5.9% and 0.08%, respectively. Therefore, when optimizing for QED or DRD2, the model would not fully extract the useful information from the training set. Since our model is trained once for all tasks (SOO and MOO), this small portion of information would potentially impact negatively for certain single property optimizations, such as QED and DRD2.

5.5.4 Multi Objective Optimization

The single objective optimization task has served as a standard benchmark in the deep-learning based molecule generation field [79, 178, 81]. However, the actual drug discovery process frequently requires balancing of multiple compound properties [162, 140]. Therefore we set up a new benchmark, multi-objective optimization (MOO). In this task, we jointly optimize three chemical properties for a given molecule. We set up the success criteria of the generated molecules in the MOO task as follows:

- $\text{sim}(X, Y) \geq 0.4$

Method	Success Rate	
	All Samples (2365)	Sub Samples (50)
VJTNN	3.56%	4.00%
MolDQN	-	0.00%
Proposed	**6.98%**	**6.00%**

Table 5.3: Multi objective optimization performance comparison.

- PlogP improvement is at least 1.0
- QED value is at least 0.9
- DRD2 value is over 0.5

To create the development set of this task, we merge all three different development sets provided by VJTNN, consisting of 1,038 molecules. Among those molecules, we exclude any molecules that already satisfy either the QED or DRD2 criteria. After this filtering, the final development set contains 985 molecules. We perform the same procedure for the test set, which reduces the number of molecules from 2,452 to 2,365.

Baselines: For this task, we include the top two baselines (MolDQN and VJTNN) from the SOO task. While MolDQN can perform the MOO task by simply modifying the reward function, VJTNN can't perform as it is because it is designed for a single property optimization. Here are how we prepare those baselines for the MOO task.

- **MolDQN**: The reward function of MolDQN for this task is defined as

$$
\begin{aligned}
r = & \frac{1}{8}\mathbb{1}(sim(X,Y) \geq 0.4) \\
& + \frac{1}{8}\mathbb{1}(P\log P(Y) - P\log P(X) \geq 1.0) \\
& + \frac{1}{8}\mathbb{1}(QED \geq 0.9) + \frac{1}{8}\mathbb{1}(DRD2 > 0.5) \\
& + \frac{1}{8}\mathrm{sim}(X,Y) + \frac{1}{8}P\log P(Y) + \frac{1}{8}\mathrm{QED}(Y) + \frac{1}{8}\mathrm{DRD2}(Y)
\end{aligned}
$$

The first four terms represent the exact goal of the task, and the last four terms provide continuous information about the goals. Since one MolDQN

model should be trained for each test sample and it requires significant time, evaluating all 2,365 samples requires more than three months with a 96-CPUs server. Therefore, we sub-sample the test set into 50 while preserving the original distribution[3] and we use it for the proxy evaluation of MolDQN. For each input, we generate 60 samples[4] after training the model (with exploration rate set to zero) and report success if at least one of them satisfies the success criteria defined above.

- **VJTNN**: We sequentially optimize an input molecule using three trained models from VJTNN (models for PlogP, QED, and DRD2). Firstly, the PlogP model generates 20 molecules for an input molecule. We select the most similar molecules that satisfy PlogP criteria. Then, for this selected molecule, the QED model generates another 20 molecules that optimize QED values. We again select the most similar molecules that satisfy PlogP and QED criteria. Finally, the DRD2 model generates 20 molecules and we report success if any of them satisfies the success criteria.

Implementation Details: We use the same configuration and the same training sets used in the previous task, SOO. We select the best model based on the development set (985 samples). When predicting a new molecule, we sample 60 molecules per input for a fair comparison to the baselines. In this task, we use the desired property vector of $\{X_{P\log P}, 0.0, 0.0\}$ and the sampled offset parameters, $\alpha = \{1.0, 2.0, 3.0\}$, $\beta = \{0.91, 0.94, 0.97, 1.0\}$, and $\gamma = \{0.51, 0.6, 0.7, 0.8, 0.9\}$.

Result: We only compare VJTNN for all samples due to the infeasible running time of MolDQN as mentioned above. As Table 5.3 shows, the proposed model is

[3]When sub-sampling, we tried to preserve the proportion of the PlogP values because other properties are already filtered by the corresponding thresholds. Not only that, to best get unbiased samples, we vary random seeds and select one that the success rate of VJTNN and the proposed method can be approximately matched with the corresponding all-sample results.

[4]We relaxed the 60-sample-condition and inspected all states (molecules) during the training because the success rate was 0% at the 50-test set. However, the success rate remains the same as 0% even if we inspect all molecules.

Method				Success Rate	±
VJTNN				3.56	-3.42
	PNet	SNet	MBS		
Proposed	☑	☑	☑	**6.98**	-
	☑	☑	☐	6.72	-0.26
	☐	☑	☑	6.77	-0.21
	☑	☐	☑	6.26	-0.72
	☐	☐	☑	5.33	-1.65
	☐	☐	☐	5.33	-1.65

Table 5.4: Ablation study. MBS is modified beam search, PNet is PropNet, and SNet is SimNet.

almost two times more successful in this task. The sub-sample experiment shows similar performance for VJTNN and ours, while MolDQN is not able to generate any successful samples.

5.5.5 Ablation Study

To illustrate the effect of the two constraint networks and the modified beam search, we present the result of the ablation study in Table 5.4. We use the MOO task for this comparison, and the result of VJTNN is also included for the reference. It's worthwhile to note that the proposed model without any constraint networks and the modified beam search still outperforms VJTNN by 1.77% point. The component with the biggest contribution is SimNet that improves the performance by 0.72% point from the model without it. Another interesting thing is the success rates of the last two models in Table 5.4 are identical. The possible explanation is that if a model is trained without any constraint networks, the neurons generating candidate molecules could not properly convey any information about similarity and properties that can be exploited in the modified beam search.

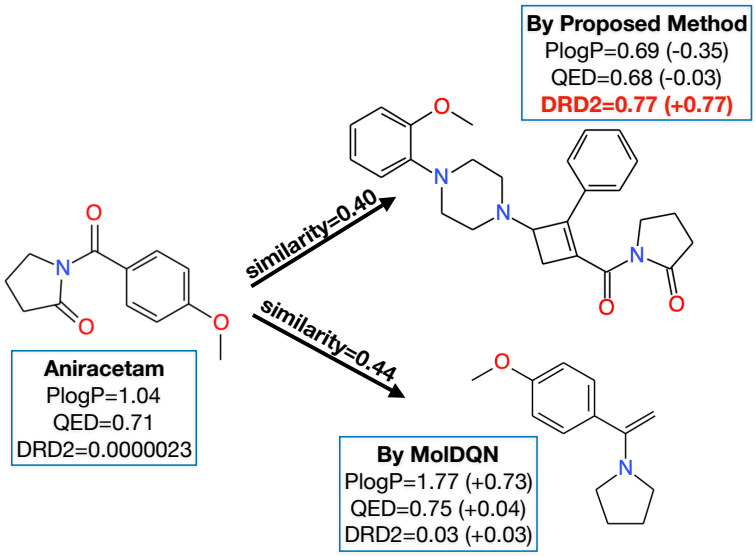


Figure 5.4: A case study: The Aniracetam optimization task to improve DRD2 score. The molecule produced by the proposed model achieved the better DRD2 score than the molecule optimized by MolDQN[5].

5.6 Case Study

We performed a case study using an actual drug that is under the experimental stage targeting Dopamine D2 receptor (DRD2). From Drugbank[6], we first enlist all DRD2 targeting drugs that are in either experimental or investigational stages. Among these 28 drugs, we select the lowest DRD2 scored drug, named Aniracetam[7] for this study. The goal is to improve DRD2 score with minimum perturbation of other properties.

Baselines: Since one VJTNN model optimizes one property, we just run the DRD2 VJTNN model trained by Jin et al. [81] by feeding Aniracetam. For MolDQN, the reward function becomes simpler as follows:

$$r = \frac{1}{2}\mathbb{1}(sim(X, Y) \geq 0.4) + \frac{1}{2}\text{DRD2}(Y)$$

Implementation Details: We use the same configuration and the same training

[6] `https://www.drugbank.ca/`
[7] Aniracetam: COC1=CC=C(C=C1)C(=O)N1CCCC1=O

sets used in the two previous tasks, SOO and MOO. We select the best model based on the development set of DRD2 task provided by VJTNN. In this task, we use the desired property vector of $\{X_{P\log P}, 0.0, 0.0\}$ and the sampled offset parameters, $\alpha = \{0.0\}$, $\beta = \{-0.1, -0.05, 0.0, 0.05, 0.1\}$, and $\gamma = \{0.6, 0.7, 0.8, 0.9\}$. We allow 20 generated samples for the proposed method and VJTNN, while we evaluate all states of MolDQN samples.

Result: In Figure 5.4, we compare the molecules generated by MolDQN[8] and ours[9], excluding the result of VJTNN, because VJTNN didn't generate valid ($sim(X, Y) \geq 0.4$) molecules. In terms of the predicted DRD2 scores, our molecule reached 0.77 whereas MolDQN's molecule only recorded 0.03. For the other two properties which should be unchanged, our molecule seems to be stable with changes in PlogP by -0.35 and QED by -0.03 when compared with the MolDQN molecule that showed larger changes especially in PlogP. Although one case study cannot prove the general superiority of the proposed model, the proposed model consistently outperforms other baselines in all benchmarks (SOO, MOO, and the case study).

5.7 Discussion

This paper proposes a new controlled molecule generation model using the self-attention based molecule translation model and two constraint networks. We pre-train and transfer the weights of the two constraint networks so that they can effectively regulate the output molecules. Not only that, we present a new beam search algorithm using these networks. Experimental results show that the proposed model outperforms all other baseline approaches in both single-objective optimization and multi-objective optimization by a large margin. Moreover, the case study using an actual experimental drug shows the practicality of the proposed model. In the ablation

[8]MolDQN Molecule: C=C(c1ccc(OC)cc1)N1CCCC1
[9]Proposed Molecule: COc1ccccc1N1CCN(C2CC(C(=O)N3CCCC3=O)=C2c2ccccc2)CC1

study, we present how each sub-unit contributes to model performance.

5.8 Contribution

In this project, my contributions are as follows.

- Designed the study with Dr. Ho.
- (solely) Developed the algorithm
- Ran experiments with Park
- Analyzed the results with Dr. Ho

6

Conclusion

The availability of well-curated biological and chemical datasets and appropriate computational power has motivated this dissertation work, deep learning approaches in computerized drug discovery. Although many machine learning based attempts have shown its potential compared to the conventional drug discovery methodology, they have been limited in modeling complex and large datasets effectively.

Inspired by many successes of deep learning based methods in other domains, this dissertation investigates the deep learning approaches for four sub-tasks in the drug discovery pipelines. In particular, we focus on two parts: target identification (Wx and CWx) and molecule discovery (MT-DTI and CMG).

Wx is a disease-related target identification model that selects a succinct set of potential target using an end-to-end feature selection neural network. Wx provides a simple way of identifying biological targets while previous methods rely on a long pipeline of machine learning methods and human labors. In addition to the simplicity of use, the results on various cancer classification tasks illustrate the superiority of the proposed method. Moreover, some of the genes we found using Wx matches those found to be important in recent studies.

CWx is a disease-related target identification model that is based on Wx to

predict patient prognosis with a small set of features. CWx outperforms the other methods by effectively training the algorithm from an easy to hard problem. In addition, CWx has a linear execution time to complete the feature selection steps depending on the number of samples. While the information theoretical-based feature selection algorithms take longer to finish the feature selection procedure. Not only that, CWx found genes have been confirmed that they contribute to the survival by other studies.

MT-DTI is a molecule transformer based drug target interaction prediction model that adapts and modifies the self-attention mechanism, which is pre-trained using publicly available big data of compounds. Although this big dataset is not directly related to the downstream tasks, the proposed pre-training is shown to be useful in DTI tasks. Experimental results show that our model outperforms all other existing methods with respect to four evaluation metrics. Moreover, the first case study of finding drug candidates targeting a cancer protein (EGFR) shows that our method successfully enlists all of the existing EGFR drugs in top-30 promising candidates. In addition, the antiviral drug discovery case study shows that the proposed method can produce useful results in a very short time.

CMG is a controlled molecule generation model that is based on the Transformer model with two constraint networks. CMG successfully adapts self-attention based translation model in the controlled molecule optimization task using a desired property vector. Not only that, we further improve the model by adding two contraint networks that predict necessary characteristics of the generated molecules. We also present a new beam search algorithm using these constraint networks. Experimental results show that the proposed model outperforms all other baseline approaches in both single-objective optimization and multi-objective optimization by a large margin. Moreover, the case study using an actual experimental drug shows the practicality of the proposed model.

7

Future Direction

There are several ways to improve and extend the proposed methods.

Interpretable DTI: In MT-DTI, we present the best DTI model, however, it is a black box model that doesn't provide precise binding pocket information, which could be valuable in other tasks of the drug discovery. By successful employing an appropriate attention mechanism to the DTI model, we can develop better and interpretable DTI models.

Protein representation: This dissertation has only explored the molecule representation via deep networks. Therefore, one possible direction is a new protein representation using deep networks. We only used the amino acids sequence for a protein, however, it actually forms three dimensional structures which contain much more information than a sequence. If we can effectively represent a protein using a predicted 3D structure, then the performance of downstream tasks could be improved.

Protein protein interaction: Once we have the better protein representation, we can apply the new representation to the protein protein interaction (PPI) task. Since a protein itself can be served as a drug, if we can understand complex PPI among many protein, it could be beneficial in developing antibody based drug development.

Bibliography

[1] Thaddeus Allen, Minke van Tuyl, Pratibha Iyengar, Serge Jothy, Martin Post, Ming-Sound Tsao, and Corrinne G Lobe. Grg1 acts as a lung-specific oncogene in a transgenic mouse model. *Cancer research*, 66(3):1294–1301, 2006.

[2] Zaccary Alperstein, Artem Cherkasov, and Jason Tyler Rolfe. All smiles variational autoencoder. *arXiv preprint arXiv:1905.13343*, 2019.

[3] Ali Anaissi, Paul J Kennedy, Madhu Goyal, and Daniel R Catchpoole. A balanced iterative random forest for gene selection from microarray data. *BMC bioinformatics*, 14(1):261, 2013.

[4] Amy C Anderson. The process of structure-based drug design. *Chemistry & biology*, 10(9):787–797, 2003.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Robert D Barber, Dan W Harmer, Robert A Coleman, and Brian J Clark. Gapdh as a housekeeping gene: analysis of gapdh mrna expression in a panel of 72 human tissues. *Physiological genomics*, 21(3):389–395, 2005.

[7] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

[8] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90, 2012.

[9] Esben Jannik Bjerrum and Richard Threlfall. Molecular generation with recurrent neural networks (rnns). *arXiv preprint arXiv:1705.04612*, 2017.

[10] Kevin Bleakley and Yoshihiro Yamanishi. Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*, 25(18):2397–2403, 2009.

[11] Paul C Boutros, Suzanne K Lau, Melania Pintilie, Ni Liu, Frances A Shepherd, Sandy D Der, Ming-Sound Tsao, Linda Z Penn, and Igor Jurisica. Prognostic gene signatures for non-small-cell lung cancer. *Proceedings of the National Academy of Sciences*, 106(8):2824–2828, 2009.

[12] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424, 2018.

[13] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[14] Mario Brosch, Kathrin Kattler, Alexander Herrmann, Witigo von Schönfels, Karl Nordström, Daniel Seehofer, Georg Damm, Thomas Becker, Sebastian Zeissig, Sophie Nehring, et al. Epigenomic map of human liver reveals principles of zonated morphogenic and metabolic control. *Nature communications*, 9(1): 1–11, 2018.

[15] Dong-Sheng Cao, Shao Liu, Qing-Song Xu, Hong-Mei Lu, Jian-Hua Huang, Qian-Nan Hu, and Yi-Zeng Liang. Large-scale prediction of drug–target in-

teractions using protein sequences and drug topological structures. *Analytica chimica acta*, 752:1–10, 2012.

[16] Dong-Sheng Cao, Liu-Xia Zhang, Gui-Shan Tan, Zheng Xiang, Wen-Bin Zeng, Qing-Song Xu, and Alex F Chen. Computational prediction of drug-target interactions using chemical, biological, and network features. *Molecular informatics*, 33(10):669–681, 2014.

[17] J Caradec, N Sirab, D Revaud, C Keumeugni, and S Loric. Is gapdh a relevant housekeeping gene for normalisation in colorectal cancer experiments? *British journal of cancer*, 103(9):1475, 2010.

[18] Josh J Carlson and Joshua A Roth. The impact of the oncotype dx breast cancer assay in clinical practice: a systematic review and meta-analysis. *Breast cancer research and treatment*, 141(1):13–22, 2013.

[19] Joy A Cavagnaro. Preclinical safety evaluation of biotechnology-derived pharmaceuticals. *Nature Reviews Drug Discovery*, 1(6):469–475, 2002.

[20] Dave Cavanagh. Coronavirus avian infectious bronchitis virus. *Veterinary research*, 38(2):281–297, 2007.

[21] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2 (3):1–27, 2011.

[22] Hsuan-Yu Chen, Sung-Liang Yu, Chun-Houh Chen, Gee-Chen Chang, Chih-Yi Chen, Ang Yuan, Chiou-Ling Cheng, Chien-Hsun Wang, Harn-Jing Terng, Shu-Fang Kao, et al. A five-gene signature and clinical outcome in non–small-cell lung cancer. *New England Journal of Medicine*, 356(1):11–20, 2007.

[23] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[24] Frederic Chibon. Cancer gene expression signatures–the rise and fall? *European journal of cancer*, 49(8):2000–2009, 2013.

[25] Hsiu-Ling Chou, Chung-Tay Yao, Sui-Lun Su, Chia-Yi Lee, Kuang-Yu Hu, Harn-Jing Terng, Yun-Wen Shih, Yu-Tien Chang, Yu-Fen Lu, Chi-Wen Chang, et al. Gene expression profiling of breast cancer survivability by pooled cdna microarray analysis using logistic regression, artificial neural networks and decision trees. *BMC bioinformatics*, 14(1):100, 2013.

[26] CM Chu, VCC Cheng, IFN Hung, MML Wong, KH Chan, KS Chan, RYT Kao, LLM Poon, CLP Wong, Y Guan, et al. Role of lopinavir/ritonavir in the treatment of sars: initial virological and clinical findings. *Thorax*, 59(3): 252–256, 2004.

[27] Murat Can Cobanoglu, Chang Liu, Feizhuo Hu, Zoltán N Oltvai, and Ivet Bahar. Predicting drug–target interactions using probabilistic matrix factorization. *Journal of chemical information and modeling*, 53(12):3399–3409, 2013.

[28] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[29] Juliana Costa-Silva, Douglas Domingues, and Fabricio Martins Lopes. Rna-seq differential expression analysis: An extended review and a software tool. *PloS one*, 12(12), 2017.

[30] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[31] David R Cox. Regression models and life-tables. In *Breakthroughs in statistics*, pages 527–541. Springer, 1992.

[32] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711, 2016.

[33] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786*, 2018.

[34] Andrew Dalke, Jerome Hert, and Christian Kramer. mmpdb: An open-source matched molecular pair platform for large multiproperty data sets. *Journal of chemical information and modeling*, 58(5):902–910, 2018.

[35] Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29 (11):1046, 2011.

[36] Mohammad Reza Dayer, Sara Taleb-Gassabi, and Mohammad Saaid Dayer. Lopinavir; a potent drug against coronavirus infection: Insight from molecular docking study. *Archives of Clinical Infectious Diseases*, 12(4), 2017.

[37] María de Guadalupe Chávez-López, Violeta Zúñiga-García, Elisabeth Hernández-Gallegos, Eunice Vera, Carmen Alexandra Chasiquiza-Anchatuña, Marco Viteri-Yánez, Janet Sanchez-Ramos, Efraín Garrido, and Javier Camacho. The combination astemizole–gefitinib as a potential therapy for human lung cancer. *OncoTargets and therapy*, 10:5795, 2017.

[38] Emmie de Wit, Neeltje van Doremalen, Darryl Falzarano, and Vincent J Mun-

ster. Sars and mers: recent insights into emerging coronaviruses. *Nature Reviews Microbiology*, 14(8):523, 2016.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[40] Glynn Dennis, Brad T Sherman, Douglas A Hosack, Jun Yang, Wei Gao, H Clifford Lane, and Richard A Lempicki. David: database for annotation, visualization, and integrated discovery. *Genome biology*, 4(9):R60, 2003.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Michael Dickson and Jean Paul Gagnon. The cost of new drug discovery and development. *Discovery medicine*, 4(22):172–179, 2009.

[43] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. Innovation in the pharmaceutical industry: new estimates of r&d costs. *Journal of health economics*, 47:20–33, 2016.

[44] Alexander G Dossetter, Edward J Griffen, and Andrew G Leach. Matched molecular pair analysis in drug discovery. *Drug Discovery Today*, 18(15-16): 724–731, 2013.

[45] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[46] Eli Eisenberg and Erez Y Levanon. Human housekeeping genes, revisited. *TRENDS in Genetics*, 29(10):569–574, 2013.

[47] Anne-Marie Ellegaard, Christian Dehlendorff, Anna C Vind, Atul Anand, Luise Cederkvist, Nikolaj HT Petersen, Jesper Nylandsted, Jan Stenvang, Anders Mellemgaard, Kell Østerlind, et al. Repurposing cationic amphiphilic antihistamines for cancer treatment. *EBioMedicine*, 9:130–139, 2016.

[48] Peter Ertl, Richard Lewis, Eric Martin, and Valery Polyakov. In silico generation of novel, drug-like chemical matter using the lstm neural network. *arXiv preprint arXiv:1712.07449*, 2017.

[49] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[50] Francesca Finotello and Barbara Di Camillo. Measuring differential gene expression with rna-seq: challenges and strategies for data analysis. *Briefings in functional genomics*, 14(2):130–142, 2015.

[51] Mario F Fraga, Maria Berdasco, Esteban Ballestar, Santiago Ropero, Pilar Lopez-Nieva, Lidia Lopez-Serra, José I Martín-Subero, Maria J Calasanz, Isabel Lopez de Silanes, Fernando Setien, et al. Epigenetic inactivation of the groucho homologue gene tle1 in hematologic malignancies. *Cancer Research*, 68 (11):4116–4122, 2008.

[52] Pierre Frères, Stéphane Wenric, Meriem Boukerroucha, Corinne Fasquelle, Jérôme Thiry, Nicolas Bovy, Ingrid Struman, Pierre Geurts, Joëlle Collignon, Hélène Schroeder, et al. Circulating microrna-based screening tool for breast cancer. *Oncotarget*, 7(5):5416, 2016.

[53] Kyle Yingkai Gao, Achille Fokoue, Heng Luo, Arun Iyengar, Sanjoy Dey, and

Ping Zhang. Interpretable drug target prediction using deep neural representation. In *IJCAI*, pages 3371–3377, 2018.

[54] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[55] Asta Gindulyte, Benjamin A Shoemaker, Bo Yu, Jia He, Jian Zhang, Jie Chen, Leonid Zaslavsky, Paul A Thiessen, Qingliang Li, Siqian He, Sunghwan Kim, Tiejun Cheng, and Evan E Bolton. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 10 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1033.

[56] Eric M Glare, Maja Divjak, MJ Bailey, and E Haydn Walters. β-actin and gapdh housekeeping gene expression in asthmatic airways is variable and not suitable for normalising mrna levels. *Thorax*, 57(9):765–770, 2002.

[57] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[58] Mehmet Gönen. Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310, 2012.

[59] Mithat Gönen and Glenn Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.

[60] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333, 2016.

[61] Susan M Greaves, Kathleen Brown, Edward B Garon, and Bonnie L Garon. The new staging system for lung cancer: imaging and clinical implications. *Journal of thoracic imaging*, 26(2):119–131, 2011.

[62] Ed Griffen, Andrew G Leach, Graeme R Robb, and Daniel J Warner. Matched molecular pairs as a medicinal chemistry tool: miniperspective. *Journal of medicinal chemistry*, 54(22):7739–7750, 2011.

[63] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[64] Gregory R Hart, David A Roffman, Roy Decker, and Jun Deng. A multi-parameterized artificial neural network for lung cancer risk prediction. *PloS one*, 13(10), 2018.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[66] Tong He, Marten Heidemeyer, Fuqiang Ban, Artem Cherkasov, and Martin Ester. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of cheminformatics*, 9(1): 24, 2017.

[67] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint arXiv:1606.08415*, 2016.

[68] Roy S Herbst. Review of epidermal growth factor receptor biology. *International Journal of Radiation Oncology* Biology* Physics*, 59(2):S21–S26, 2004.

[69] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[70] JS Hopkins. U.s. drugmakers ship therapies to china, seeking to treat coronavirus. *The Wall Street Journal.*, 2020.

[71] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[72] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017.

[73] Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *arXiv preprint arXiv:1811.06965*, 2018.

[74] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

[75] Navdeep Jaitly, Patrick Nguyen, Andrew Senior, and Vincent Vanhoucke. Application of pretrained deep neural networks to large vocabulary speech recog-

nition. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[76] Stanislaw Jastrzebski, Damian Leśniak, and Wojciech Marian Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.

[77] Miles F Jefferson, Neil Pendleton, Sam B Lucas, and Michael A Horan. Comparison of a genetic algorithm neural network with logistic regression for predicting outcome after surgery for patients with nonsmall cell lung carcinoma. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 79(7): 1338–1342, 1997.

[78] Shih Sheng Jiang, Wen-Tsen Fang, Ya-Hsiue Hou, Shiu-Feng Huang, B Linju Yen, Junn-Liang Chang, Shih-Miao Li, Hui-Ping Liu, Ying-Lan Liu, Chih-Ting Huang, et al. Upregulation of sox9 in lung adenocarcinoma and its involvement in the regulation of cell growth and tumorigenicity. *Clinical Cancer Research*, 16(17):4363–4373, 2010.

[79] Wengong Jin, Connor Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. In *Advances in Neural Information Processing Systems*, pages 2607–2616, 2017.

[80] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018.

[81] Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *ICLR*, 2019.

[82] Subha Kalyaanamoorthy and Yi-Ping Phoebe Chen. Structure-based drug design to augment hit discovery. *Drug discovery today*, 16(17-18):831–839, 2011.

[83] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1): D1102–D1109, 2018.

[84] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[85] Peter Kirkpatrick and Clare Ellis. Chemical space, 2004.

[86] Daniel C Koboldt, Karyn Meltz Steinberg, David E Larson, Richard K Wilson, and Elaine R Mardis. The next-generation sequencing revolution and its impact on genomics. *Cell*, 155(1):27–38, 2013.

[87] Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer, 1994.

[88] Thijs Kuiken, Ron AM Fouchier, Martin Schutten, Guus F Rimmelzwaan, Geert Van Amerongen, Debby van Riel, Jon D Laman, Ton de Jong, Gerard van Doornum, Wilina Lim, et al. Newly discovered coronavirus as the primary cause of severe acute respiratory syndrome. *The Lancet*, 362(9380):263–270, 2003.

[89] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1945–1954. JMLR. org, 2017.

[90] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[91] Qun Li, Xuhua Guan, Peng Wu, Xiaoye Wang, Lei Zhou, Yeqing Tong, Ruiqi Ren, Kathy SM Leung, Eric HY Lau, Jessica Y Wong, et al. Early transmission dynamics in wuhan, china, of novel coronavirus–infected pneumonia. *New England Journal of Medicine*, 2020.

[92] Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):33, 2018.

[93] Yvonne Xinyi Lim, Yan Ling Ng, James P Tam, and Ding Xiang Liu. Human coronaviruses: a review of virus–host interactions. *Diseases*, 4(3):26, 2016.

[94] David J Lipman and William R Pearson. Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435–1441, 1985.

[95] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[96] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 339–348. AUAI Press, 2009.

[97] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

[98] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15 (12):550, 2014.

[99] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, pages 436–440, 2013.

[100] Heng Luo, William Mattes, Donna L Mendrick, and Huixiao Hong. Molecular docking for identification of potential targets for drug repurposing. *Current topics in medicinal chemistry*, 16(30):3636–3645, 2016.

[101] Yutaka Maeda, Vrushank Davé, and Jeffrey A Whitsett. Transcriptional control of lung morphogenesis. *Physiological reviews*, 87(1):219–244, 2007.

[102] Raghvendra Mall, Luigi Cerulo, Luciano Garofano, Veronique Frattini, Khalid Kunji, Halima Bensmail, Thais S Sabedot, Houtan Noushmehr, Anna Lasorella, Antonio Iavarone, et al. Rgbm: regularized gradient boosting machines for identification of the transcriptional regulators of discrete glioma subtypes. *Nucleic acids research*, 46(7):e39–e39, 2018.

[103] Alvaro Mateos, Joaquın Dopazo, Ronald Jansen, Yuhai Tu, Mark Gerstein, and Gustavo Stolovitzky. Systematic learning of gene functional classes from dna array expression data by using multilayer perceptrons. *Genome Research*, 12 (11):1703–1715, 2002.

[104] Patrick J Medina and Susan Goodin. Lapatinib: a dual inhibitor of human epidermal growth factor receptor tyrosine kinases. *Clinical therapeutics*, 30(8): 1426–1447, 2008.

[105] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al. Speech understanding systems: Report of a steering committee. *Artificial Intelligence*, 9(3):307–316, 1977.

[106] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks*

for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468), pages 41–48. Ieee, 1999.

[107] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[108] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[109] Joseph Monforte and Steve McPhail. Strategy for gene expression-based biomarker discovery. *BioTechniques*, 38(S4):S25–S29, 2005.

[110] Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan. Trace ratio criterion for feature selection. In *AAAI*, volume 2, pages 671–676, 2008.

[111] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

[112] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

[113] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. A comparative study of smiles-based compound similarity functions for drug-target interaction prediction. *BMC bioinformatics*, 17(1):128, 2016.

[114] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[115] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[116] Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyawar, Agnieszka Szwajda, Jing Tang, and Tero Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337, 2014.

[117] George Papadatos, Anna Gaulton, Anne Hersey, and John P Overington. Activity, assay and target data curation and quality in the chembl database. *Journal of computer-aided molecular design*, 29(9):885–896, 2015.

[118] Sungsoo Park, Bonggun Shin, Won Sang Shim, Yoonjung Choi, Kilsoo Kang, and Keunsoo Kang. Wx: a neural network-based feature selection algorithm for transcriptomic data. *Scientific reports*, 9(1):1–9, 2019.

[119] Peter Peduzzi, John Concato, Elizabeth Kemper, Theodore R Holford, and Alvan R Feinstein. A simulation study of the number of events per variable in logistic regression analysis. *Journal of clinical epidemiology*, 49(12):1373–1379, 1996.

[120] Li Peng, Xiu Wu Bian, Chuan Xu, Guang Ming Wang, Qing You Xia, Qing Xiong, et al. Large-scale rna-seq transcriptome analysis of 4043 cancers and 548 normal tissue controls across 12 tcga cancer types. *Scientific reports*, 5:13413, 2015.

[121] Yasset Perez-Riverol, Max Kuhn, Juan Antonio Vizcaino, Marc-Phillip Hitz, and Enrique Audain. Accurate and fast feature selection workflow for high-dimensional omics data. *PloS one*, 12(12):e0189875, 2017.

[122] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[123] Yi-Ping Phoebe Chen and Feng Chen. Identifying targets for drug discovery using bioinformatics. *Expert opinion on therapeutic targets*, 12(4):383–389, 2008.

[124] Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27(8):675–679, 2013.

[125] Partha Pratim Roy, Somnath Paul, Indrani Mitra, and Kunal Roy. On two novel parameters for validation of predictive qsar models. *Molecules*, 14(5): 1660–1701, 2009.

[126] Andrey Ptitsyn, Matthew Hulver, William Cefalu, David York, and Steven R Smith. Unsupervised clustering of gene expression data points at hypoxia as possible trigger for metabolic syndrome. *BMC genomics*, 7(1):318, 2006.

[127] Robert A Quinn, Louis-Felix Nothias, Oliver Vining, Michael Meehan, Eduardo Esquenazi, and Pieter C Dorrestein. Molecular networking as a drug discovery, drug metabolism, and precision medicine strategy. *Trends in pharmacological sciences*, 38(2):143–154, 2017.

[128] Sridhar Ramaswamy and Charles M Perou. Dna microarrays in breast cancer: the promise of personalised medicine. *The Lancet*, 361(9369):1576–1576, 2003.

[129] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4780–4789, 2019.

[130] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[131] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[132] Florian Rohart, Benoît Gautier, Amrit Singh, and Kim-Anh Lê Cao. mixomics: An r package for omics feature selection and multiple data integration. *PLoS computational biology*, 13(11):e1005752, 2017.

[133] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15, 2015.

[134] Dimitrios H Roukos. Next-generation, genome sequencing-based biomarkers: concerns and challenges for medical practice. *Biomarkers in medicine*, 4(4): 583–586, 2010.

[135] Kunal Roy, Pratim Chakraborty, Indrani Mitra, Probir Kumar Ojha, Supratik Kar, and Rudra Narayan Das. Some case studies on application of rm2 metrics for judging quality of quantitative structure–activity relationship predictions: emphasis on scaling of response data. *Journal of computational chemistry*, 34 (12):1071–1082, 2013.

[136] Bidisha Samanta, Abir De, Niloy Ganguly, and Manuel Gomez-Rodriguez. Designing random graph models using variational autoencoders with applications to chemical design. *arXiv preprint arXiv:1802.05283*, 2018.

[137] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L Guimaraes, and Alán Aspuru-Guzik. Optimizing distributions over molecular space. an objective-

reinforced generative adversarial network for inverse-design chemistry (organic). 2017.

[138] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131, 2018.

[139] Sung Wook Seo, Hyewon Lee, Hyun-Il Lee, and Han-Soo Kim. The role of tle1 in synovial sarcoma. *Journal of orthopaedic research*, 29(7):1131–1136, 2011.

[140] Veerabahu Shanmugasundaram, Liying Zhang, Shilva Kayastha, Antonio de la Vega de Leon, Dilyana Dimova, and Jurgen Bajorath. Monitoring the progression of structure–activity relationship information during lead optimization. *Journal of medicinal chemistry*, 59(9):4235–4244, 2016.

[141] Bonggun Shin, Falgun H Chokshi, Timothy Lee, and Jinho D Choi. Classification of radiology reports using neural attention models. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4363–4370. IEEE, 2017.

[142] Bonggun Shin, Sungsoo Park, Ji Hyung Hong, Ho Jung An, Sang Hoon Chun, Kilsoo Kang, Young-Ho Ahn, Yoon Ho Ko, and Keunsoo Kang. Cascaded wx: a novel prognosis-related feature selection framework in human lung adenocarcinoma transcriptomes. *Frontiers in genetics*, 10:662, 2019.

[143] Bonggun Shin, Sungsoo Park, Keunsoo Kang, and Joyce C. Ho. Self-attention based molecule representation for predicting drug-target interaction. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 230–248. PMLR, 09–10 Aug 2019.

[144] Bonggun Shin, Sungsoo Park, Keunsoo Kang, and Joyce C Ho. Self-attention

based molecule representation for predicting drug-target interaction. *MLHC*, 2019.

[145] Sara Sigismund, Daniele Avanzato, and Letizia Lanzetti. Emerging functions of the egfr in cancer. *Molecular oncology*, 12(1):3–20, 2018.

[146] Kavleen Sikand, Jagjit Singh, Jey Sabith Ebron, and Girish C Shukla. Housekeeping gene selection advisory: glyceraldehyde-3-phosphate dehydrogenase (gapdh) and β-actin are targets of mir-644a. *PloS one*, 7(10):e47510, 2012.

[147] Richard B Silverman and Mark W Holladay. *The organic chemistry of drug design and drug action*. Academic press, 2014.

[148] Marcin Skrzypski, Ewa Jassem, Miquel Taron, Jose Javier Sanchez, Pedro Mendez, Witold Rzyman, Grazyna Gulida, Dan Raz, David Jablons, Mariano Provencio, et al. Three-gene expression signature predicts survival in early-stage squamous cell carcinoma of the lung. *Clinical Cancer Research*, 14(15): 4794–4799, 2008.

[149] Jean-Charles Soria, Yuichiro Ohe, Johan Vansteenkiste, Thanyanan Reungwetwattana, Busyamas Chewaskulyong, Ki Hyeong Lee, Arunee Dechaphunkul, Fumio Imamura, Naoyuki Nogami, Takayasu Kurata, et al. Osimertinib in untreated egfr-mutated advanced non–small-cell lung cancer. *New England journal of medicine*, 378(2):113–125, 2018.

[150] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

[151] Tuomas Tammela, Francisco J Sanchez-Rivera, Naniye Malli Cetinbas, Katherine Wu, Nikhil S Joshi, Katja Helenius, Yoona Park, Roxana Azimi, Natanya R Kerper, R Alexander Wesselhoeft, et al. A wnt-producing niche drives prolif-

erative potential and progression in lung adenocarcinoma. *Nature*, 545(7654): 355, 2017.

[152] Jing Tang, Agnieszka Szwajda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.

[153] ZiaurRehman Tanoli, Zaid Alam, Markus Vähä-Koskela, Balaguru Ravikumar, Alina Malyutina, Alok Jaiswal, Jing Tang, Krister Wennerberg, and Tero Aittokallio. Drug target commons 2.0: a community platform for systematic analysis of drug–target interaction profiles. *Database*, 2018, 2018.

[154] Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth, Daniel Treacy, John J Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, 352(6282):189–196, 2016.

[155] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019.

[156] Shashank Tripathi, Marie O Pohl, Yingyao Zhou, Ariel Rodriguez-Frandsen, Guojun Wang, David A Stein, Hong M Moulton, Paul DeJesus, Jianwei Che, Lubbertus CF Mulder, et al. Meta-and orthogonal integration of influenza omics data defines a role for ubr4 in virus budding. *Cell host & microbe*, 18(6): 723–735, 2015.

[157] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[158] Twan van Laarhoven, Sander B Nabuurs, and Elena Marchiori. Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, 27(21):3036–3043, 2011.

[159] Laura J Van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin Van Der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415(6871):530–536, 2002.

[160] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[161] Emma E Vincent, Douglas JE Elder, O Linda, Olivier E Pardo, Piotr Dzien, Lois Phillips, Carys Morgan, Joya Pawade, Margaret T May, Muhammad Sohail, et al. Glycogen synthase kinase 3 protein kinase activity is frequently elevated in human non-small cell lung carcinoma and supports tumour cell proliferation. *PloS one*, 9(12):e114725, 2014.

[162] Martin Vogt, Dimitar Yonchev, and Jurgen Bajorath. Computational method to evaluate progress in lead optimization. *Journal of medicinal chemistry*, 61 (23):10895–10900, 2018.

[163] Jing-Fang Wang, Dong-Qing Wei, and Kuo-Chen Chou. Pharmacogenomics and personalized use of drugs. *Current topics in medicinal chemistry*, 8(18): 1573–1579, 2008.

[164] Tianyu Wang, Boyang Li, Craig E Nelson, and Sheida Nabavi. Comparative analysis of differential gene expression analysis tools for single-cell rna sequencing data. *BMC bioinformatics*, 20(1):40, 2019.

[165] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[166] Susan R Weiss and Julian L Leibowitz. Coronavirus pathogenesis. In *Advances in virus research*, volume 81, pages 85–164. Elsevier, 2011.

[167] Stephane Wenric and Ruhollah Shemirani. Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in genetics*, 9:297, 2018.

[168] Ben S Wittner, Dennis C Sgroi, Paula D Ryan, Tako J Bruinsma, Annuska M Glas, Anitha Male, Sonika Dahiya, Karleen Habin, Rene Bernards, Daniel A Haber, et al. Analysis of the mammaprint breast cancer assay in a predominantly postmenopausal cohort. *Clinical Cancer Research*, 14(10):2988–2993, 2008.

[169] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[170] Patrick CY Woo, Yi Huang, Susanna KP Lau, and Kwok-Yung Yuen. Coronavirus genomics and bioinformatics analysis. *viruses*, 2(8):1804–1820, 2010.

[171] Wenjie Xia, Xinnian Yu, Qixing Mao, Wenying Xia, Anpeng Wang, Gaochao Dong, Bing Chen, Weidong Ma, Lin Xu, and Feng Jiang. Improvement of survival for non-small cell lung cancer over time. *OncoTargets and therapy*, 10: 4295, 2017.

[172] Nan-Nan Xie, Liang Hu, and Tai-Hui Li. Lung cancer risk prediction method based on feature selection and artificial neural network. *Asian Pac J Cancer Prev*, 15(23):10539–10542, 2014.

[173] Yang Xie, Guanghua Xiao, Kevin R Coombes, Carmen Behrens, Luisa M Solis, Gabriela Raso, Luc Girard, Heidi S Erickson, Jack Roth, John V Heymach, et al. Robust gene expression signature from formalin-fixed paraffin-embedded samples predicts prognosis of non–small-cell lung cancer patients. *Clinical Cancer Research*, 17(17):5705–5714, 2011.

[174] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.

[175] Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. Chemts: an efficient python library for de novo molecular generation. *Science and technology of advanced materials*, 18(1):972–976, 2017.

[176] Dengju Yao, Jing Yang, Xiaojuan Zhan, Xiaorong Zhan, and Zhiqiang Xie. A novel random forests-based feature selection method for microarray expression data analysis. *International journal of data mining and bioinformatics*, 13(1): 84–101, 2015.

[177] Xin Yao, Shubha Kale Ireland, Tri Pham, Brandi Temple, Renwei Chen, Madhwa HG Raj, and Hector Biliran. Tle1 promotes emt in a549 lung cancer cells through suppression of e-cadherin. *Biochemical and biophysical research communications*, 455(3-4):277–284, 2014.

[178] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, pages 6410–6421, 2018.

[179] Kun Yu, Kumaresan Ganesan, Lay Keng Tan, Mirtha Laban, Jeanie Wu, Xiao Dong Zhao, Hongmin Li, Carol Ho Wing Leung, Yansong Zhu, Chia Lin

Wei, et al. A precisely regulated gene expression cassette potently modulates metastasis and survival in multiple solid cancers. *PLoS genetics*, 4(7):e1000129, 2008.

[180] Ali M Zaki, Sander Van Boheemen, Theo M Bestebroer, Albert DME Osterhaus, and Ron AM Fouchier. Isolation of a novel coronavirus from a man with pneumonia in saudi arabia. *New England Journal of Medicine*, 367(19): 1814–1820, 2012.

[181] Mikhail Zaslavskiy, Simon Jégou, Eric W Tramel, and Gilles Wainrib. Toxicblend: Virtual screening of toxic compounds with ensemble predictors. *Computational Toxicology*, 10:81–88, 2019.

[182] Jing Zhang, Hanane Hadj-Moussa, and Kenneth B Storey. Current progress of high-throughput microrna differential expression analysis and random forest gene selection for model and non-model systems: an r implementation. *Journal of integrative bioinformatics*, 13(5):35–46, 2016.

[183] Wei Zhang, Jeremy Chien, Jeongsik Yong, and Rui Kuang. Network-based machine learning and graph theory algorithms for precision oncology. *NPJ precision oncology*, 1(1):1–15, 2017.

[184] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9 (1):1–10, 2019.

[185] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.