DISTANCE MATRICES AS PROTEIN REPRESENTATIONS

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

BY

MEHMET DINÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2022

Approval of the thesis:

DISTANCE MATRICES AS PROTEIN REPRESENTATIONS

submitted by **MEHMET DINÇ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

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

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

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

Examining Committee Members:

Prof. Dr. Hakkı Toroslu
Computer Engineering, METU _____

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

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

Date: 02.09.2022

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

Name, Surname: Mehmet Dinç

Signature :

ABSTRACT

DISTANCE MATRICES AS PROTEIN REPRESENTATIONS

Dinç, Mehmet
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. M. Volkan Atalay

SEPTEMBER 2022, 56 pages

Representing protein sequences is a crucial problem in the field of bioinformatics since any data-driven model's performance is limited by the information contained in its input features. A protein's biological function is dictated by its structure and knowing a protein's structure can potentially help predict its interactions with drug candidates or predict its Gene Ontology (GO) term. Yet, off-the-shelf protein representations do not contain such information since only a small fraction of the billions of known protein sequences have experimentally determined structures, as the cost of running such experiments is quite high. A newly introduced neural network-based structure prediction model, AlphaFold, claims to be able to predict protein structures with high accuracy. In this study, two-dimensional distance matrices generated from AlphaFold structure predictions are used as input features while modeling two different bioinformatics problems; drug-target interaction (DTI) prediction and Gene Ontology term prediction. For the DTI prediction problem, a state-of-the-art model which already uses two-dimensional protein features, is employed as a baseline. Then, the effect of distance matrices is observed through ablation studies. Moreover, the same model is adapted in order to tackle the GO prediction problem and its success is compared with off-the-shelf protein representations.

Keywords: protein distance matrices, drug-target interaction prediction, gene ontology prediction

ÖZ

UZAKLIK MATRİSLERİNİN PROTEİN REPREZANTASYONU OLARAK KULLANIMI

Dinç, Mehmet
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. M. Volkan Atalay

Eylül 2022 , 56 sayfa

Herhangi bir veriye dayalı modelin performansı, girdi özniteliklerinde yer alan bilgilerle sınırlı olduğundan, protein dizilerini temsil etmek biyoinformatik alanında çok önemli bir sorundur. Bir proteinin biyolojik işlevi yapısı tarafından belirlenir ve bir proteinin yapısını bilmek, potansiyel olarak ilaç adaylarıyla etkileşimlerini tahmin etmeye veya Gen Ontolojisi (GO) terimini tahmin etmeye yardımcı olabilir. Yine de, mevcut protein gösterimleri bu tür bilgileri içermemektedir zira bilinen milyarlarca protein dizisinin yalnızca küçük bir kısmı deneysel olarak belirlenmiş yapılara sahiptir. Bunun nedeni de bu tür deneyleri yürütmenin maliyetinin oldukça yüksek olmasıdır. Yeni tanıtılan bir sinir ağı tabanlı yapı tahmin modeli AlphaFold, protein yapılarını yüksek doğrulukla tahmin edebildiğini iddia etmektedir. Bu çalışmada, AlphaFold yapı tahminlerinden üretilen iki boyutlu uzaklık matrisleri, iki farklı biyoinformatik problemi modellenirken girdi özniteliği olarak kullanılmıştır; ilaç-hedef etkileşimi tahmini ve Gen Ontolojisi tahmini. Ilaç-hedef etkileşimi tahmin problemi için, halihazırda iki boyutlu protein özelliklerini kullanan son teknoloji bir model, temel olarak kullanılıştır. Daha sonra ablasyon çalışmaları ile uzaklık matrislerinin etkisi

gözlemlenmiştir. Ayrıca, aynı model, GO tahmin probleminin üstesinden gelmek için uyarlanmıştır ve başarısı, hazır protein temsilleriyle karşılaştırılmıştır.

Anahtar Kelimeler: protein uzaklık matrisleri, ilaç-hedef etkileşimi tahmini, gen ontolojisi tahmini

ACKNOWLEDGMENTS

Thanks to my supervisor, M. Volkan Atalay, for providing guidance and feedback throughout the last few years.

I'd also like to thank my father for his emotional support and funding for this and my previous studies.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
GO	Gene Ontology
DTI	Drug-Target Intereaction
ReLU	Rectified Linear Unit
FC	Fully Connected
VS	Virtual Screening
SMILES	Simplified Molecular-input Line-entry System
ECFP	Extended Connectivity Fingerprints
PDB	The Protein Data Bank
Å	Angstrom
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

Data-driven methods that are also called machine learning methods are becoming increasingly popular in bioinformatics similar to other various fields. Nowadays, data-driven methods are widely used in tasks such as virtual screening, and protein function prediction. These machine learning methods, especially deep learning methods which have a high number of learnable parameters, require a significant amount of data in order to perform reasonably. Each day, the number of available data in the world grows exponentially and biology is no different. With efforts such as the Human Genome Project [1], the cost of genome sequencing is drastically reduced. As of August 2022, there are 226,771,948 proteins available in UniProt [2]. However, these numbers themselves do not mean much. For a machine learning model to thrive, the information contained in the input is as important as the amount of available data. Yet, a protein sequence (also known as the primary structure) itself is simply a string of characters (amino acids), and unable to represent a lot of, possibly crucial, information about a protein. This situation is a limiting factor for the performance of machine learning algorithms used in bioinformatics. To overcome this problem, researchers proposed a plethora of different protein representations. Further information about these methods is given in Chapter 2.

Proteins function by physically interacting with other molecules inside the cell. Such protein-protein or protein-compound interactions are determined by the 3D shape of the respective proteins and compounds. Therefore, knowing a protein's 3D structure massively helps when determining its interactions. However, the 3D structure of a protein is hard to come by. Experimentally determining a protein's structure is an expensive and tedious task. Despite its importance and the recent advancements

in X-ray crystallography [3], only 25,984 proteins in UniProt have experimentally determined structures. This amounts to less than 1% of the proteins available in UniProt.

To fill this gap, many researchers proposed various methods that can predict a protein's structure from its amino acid sequence. In order to assess these methods, Critical Assessment of protein Structure Prediction (CASP) is founded in 1994 [4]. CASP is a protein structure prediction challenge taking place every two years. In the most recent CASP round, CASP14, AlphaFold 2 massively outscored its opponents and generated unexpectedly accurate predictions. According to scientists, this is a major breakthrough that would majorly impact related subfields of biology [5]. While being far from perfect, it is believed that AlphaFold's predictions might be good enough when experimentally determined structures are unavailable.

Our research was conducted with the goal of answering the question of whether these predictions could be helpful in solving bioinformatics problems or not. In order to do so, the distance matrices constructed using AlphaFold's predictions are employed to tackle two prevalent problems: the drug-target interaction (Virtual Screening) prediction and the Gene Ontology term (protein function) prediction. First, to unlock the potential of distance matrices, a convolutional network, which was already built to extract features from multi-channel 2D matrices, is adapted from an in-house publication. By running ablation studies, the significance of distance matrices and their potential to further improve existing successful methods is shown. Then, the same method is used on GO term prediction. Again, the importance of distance matrices is made evident by ablation studies. However, our method is shown to underperform when compared to different off-the-shelf methods run on the same datasets, something for which a possible explanation is given.

This thesis is structured in the following way. In Chapter 2 related work is presented. In Chapter 3, the construction of input protein feature matrices is explained, along with the details of the deep neural network architecture used in this study. The application of our method to drug-target interaction prediction is explained in Chapter 4, with a literature review on the subject. In Chapter 5, our attempt at applying our method to Gene Ontology term prediction is explained, along with a comparison to

an external method and a literature survey. Finally, In Chapter 6, this thesis is concluded with final remarks and possible future directions.

CHAPTER 2

RELATED WORK

While the distance matrices used in this study are generated from 3D structure predictions, it is not the only way of generating distance matrices for proteins with unknown 3D structures. Direct prediction of protein residue distance matrices (also referred to as distance maps) is a heavily studied subject as it might be considered as a simplified version of protein structure prediction problem [6]. Moreover, distance matrices are not the only way of representing protein features. In this chapter, literature about both the prediction of distance matrices and alternative protein representation methods is presented.

Most of the methods mentioned below use Multiple Sequence Alignment (MSA) based methods such as covariance matrix, precision matrix, etc. as input features to their prediction models. Researchers also resort to non-coevolutionary features such as position-specific scoring matrix (PSSM) and solvent accessibility.

Xu *et al.* used a neural network with several ResNet [7] blocks in order to predict protein distance matrices [8]. However, instead of using regression, they created 25 bins of 0.5Å and tackled the problem as a multi-class classification problem. Ding *et al.* took the generative modeling approach and proposed a Generative Adversarial Network (GAN) based method that is called GANProDist [9]. DeepDist, proposed by Wu *et al.*, is another method that uses deep residual convolutional networks [10]. What is significant about this work is authors trained their model by using the multi-task learning approach. DeepDist is trained to predict a real-value distance matrix and a multi-class distance matrix (where distances are binned, similar to [8]). Authors claim that this method improves the performance over only predicting the real-value distance matrix. In their work, Rahman *et al.* showed more features are not always

mean better results. According to the authors, more features bring extra noise and complicate things for the prediction model. By reducing the number of features fed to the model (compared to previous work), they achieved better predictions [11].

Protein contact maps are similar to distance maps. Instead of showing the actual distance between each residue, contact maps only show whether two residues are in contact or not. A pair of residues are considered to be in contact if their distance is below a certain threshold, usually 9Å. Contact maps are relatively easier to successfully predict and thus even more studied than distance matrices [12, 13, 14].

An amino acid substitution scoring matrix is a 20×20 matrix that represents the rates at which various amino acids in proteins are being substituted by other amino acids over time [15]. Values contained in these matrices can be used to generate 2D representations of protein sequences, as shown in Figure 3.2. Most of these matrices are available in AAIndex [16]. Saini *et al.* proposed a k-separated bigram (a pair of amino acids) technique that represents bigram transition probabilities using the position-specific scoring matrix (PSSM) [17]. The number k determines the spatial separation between the amino acids in the bigram and does not have a predefined optimum value. Pseudo amino acid composition (PAAC) [18], and its successor amphiphilic pseudo amino acid composition (APAAC) [19] are two prevalent protein representations. Both representations make use of the physicochemical properties of amino acids and their compositions.

Since protein sequences are represented as strings of letters, there are a lot of conceptual similarities -and differences- between proteins and human language [20]. This inspired bioinformaticians to adapt Natural Language Processing (NLP) methods to proteins. By using tricks like treating k-mers as words or sequences as sentences, researchers were able to use famous NLP methods like word2vec [21], doc2vec [22], BERT [23] and others [24, 25, 26, 27, 28, 29].

CHAPTER 3

PROTEIN REPRESENTATIONS AND DEEP LEARNING ARCHITECTURE

The architecture used in this study is heavily adopted from MDeePred [30]. MDeePred uses proteochemometric modelling, meaning both the protein and the compound features are fed to the system. In MDeePred, compounds are represented by circular molecular fingerprints (ECFP4). Proteins, however, are represented by multi-channel, 2D feature matrices. This made the architecture used in MDeePred suitable for this study, as the distance matrices are two-dimensional and can simply be appended as another channel for protein features.

3.1 Distance Matrices

A protein distance matrix is an $N \times N$ matrix that shows the distance between every amino acid residue pair in a protein structure, where *N* is the length of the protein sequence. A protein distance matrix is calculated by measuring the distance between C_a atoms of each residue. It is a reduced (2D) version of the actual 3D atomic structure of a protein, yet there are several advantages of using distance matrices instead of the actual 3D structures. First of all distance matrices are invariant to rotations and translations. Second, they require less computation to be processed. Finally, there exist a plethora of machine learning methods, especially from the computer vision field, that excel at handling 2D data.

Structures used in our work are acquired from "AlphaFold Protein Structure Database (AlphaFold DB)" [31], where predicted structures of whole proteomes of certain organisms, including humans are hosted. Downloaded structures are in PDB format which is a textual file format describing the three-dimensional structures of molecules

held in the Protein Data Bank. Then, freely-available PConPy [32] software is used to generate 2D distance matrices from the PDB files. Figure 3.1 visualizes this pipeline.

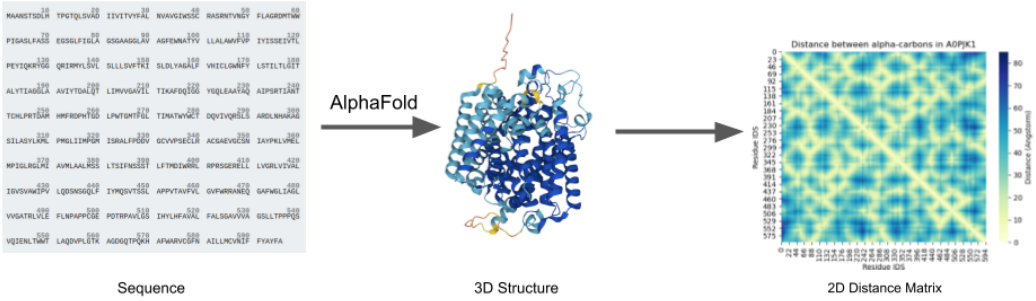

Figure 3.1: Pipeline of distance matrix generation. Using only a protein sequence, AlphaFold 2 outputs a PDB file showing the predicted 3D structure. Then, PConPy uses these 3D coordinates to create a 2D distance matrix.

3.2 Protein feature matrices taken from AAindex database

There are several different matrices used for protein representation. First of them is 2D encoding matrices, generated in order to act as a base input channel. These 2D encoding matrices do not have any chemical/physical information packed in them. Their sole purpose is to represent "each possible amino acid pair in a sequence matrix" with a unique integer. This process is show in the part 'A' of the Figure 3.2.

In order to represent the proteins using biological, chemical and physical information, four amino acid matrices are selected from the AAindex [16]. First two are SIMK990101 [33] and ZHAC000103 [34], both of which bring in structural information about the protein, shown in part 'D' of the Figure 3.2. Third one is the BLOSUM62 [35] scoring matrix that brings evolutionary information, as in part 'B' of Figure 3.2. The last of these matrices is the Grantham matrix—GRAR740104 [36]

which represents the physiochemical property differences between amino acids over composition, polarity and molecular volume, again shown in Figure 3.2 under part 'c'. These features are further summarized in table 3.1

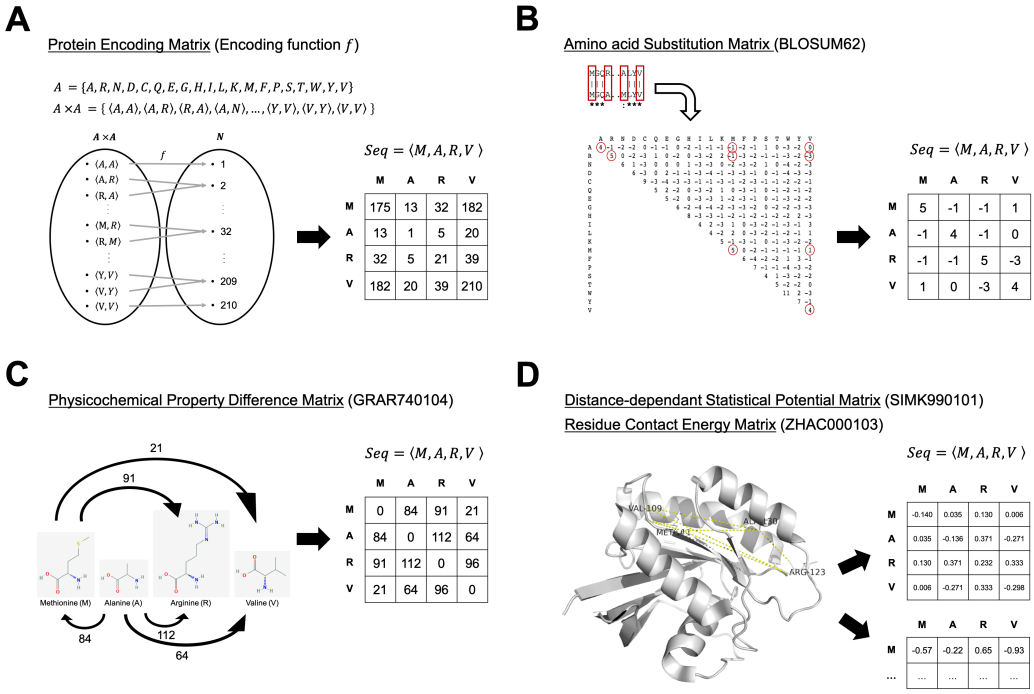

Figure 3.2: Feature matrix generation for a sample sequence "MARV". Adapted from [30]

Table 3.1: Protein Feature Matrices

AAIndex Database ID	Feature Type	Name of the Matrix
ZHAC000103	Structure Based	Environment-dependent residue contact energies
BLOSUM62	Evolutionary	Amino acid substitution
GRAR740104	Physicochemical	Chemical distance
SIMK990101	Structure-based	Distance-dependent statistical potential

Protein sequences have various lengths. The shortest protein sequence in humans is two amino acids long, whereas the longest is 35,991. However, convolutional neural networks require fixed-size inputs to function properly. Thus, protein sequences needed to be cut or padded to a certain length. When deciding on this length, there is a trade-off between information and computation time. As the length increases, the

amount of information about the protein that is captured increases. However, since the matrices are 2D, their size grows quadratically (N^2). Protein feature matrices used in this study are modified to have the shape of 500×500. Proteins that are shorter than 500 AAs are zero-padded equally from both sides, whereas the longer proteins are cropped equally from both sides as well. This process is further illustrated in Figure 3.3.

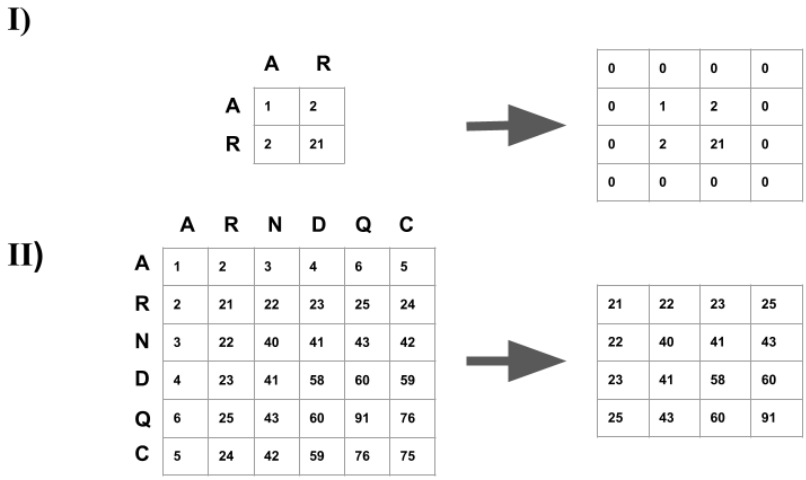

Figure 3.3: I) The sequence "AR" is zero-padded from both sides to have the shape of 4×4. II) The sequence "ARNDQC" is cropped from both sides to have the shape of 4×4

3.3 Preliminaries for Deep Learning

A fully connected(FC) layer applies a linear transformation to the input vector by multiplying it with a weight matrix.

$$y = \sum_{i=1}^{n} (x_i w_i) + b \tag{3.1}$$

where y is the output of the FC layer and the x is the input. w represents the layer's weights and b represents the bias term. Fully connected layers are usually followed

by a non-linear transformation such as the Rectified Linear Unit(ReLU) [37] which is defined as:

$$f(x) = \begin{cases} 0 & \text{if} \quad x < 0 \\ x & \text{if} \quad x \geq 0 \end{cases} \tag{3.2}$$

A convolutional layer is defined as:

$$S(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{3.3}$$

Where I is the 2D input and K is the 2D kernel with learnable parameters. S is commonly referred as the feature map. Usually, this feature map is subjected to a pooling function in order to reduce its size. Max Pooling [38] is a pooling operation that calculates the maximum value of fixed size patches in a feature map.

Inception layer apply multiple convolutions with varying filter sizes simultaneously in the same layer [39]. Aim of a inception layer is to allow neural network use the most meaningful features coming from the parallel filters, instead of selecting a single filter size and using it. Downside of inception layers is that they require significant amount of computation.

3.4 Architecture Used in DTI Prediction

The model used in this task is a pairwise-input neural network built to achieve proteochemometric modeling of the drug-target interactions. Target proteins first go through a convolutional neural network that consists of 2 blocks, wherein each block, convolution operation, ReLU activation, and max pooling is applied to the input, in that order. The output of these blocks is then fed to the Inception layer. Finally, the output of the inception module is flattened to be a 1D vector. On the other hand, compounds that are 1024 dimensional ECFP4 fingerprint vectors pass through a network with two fully connected layers, where each FC is followed by batch normalization [40] and ReLU function. After both protein and the compound passes through

their respective networks, the output of these networks is concatenated in order to have a feature vector that includes information about both the protein and the compound. This concatenated vector then passes through a network with two fully connected layers, similar to the one that compounds pass through. Figure 3.4 shows the overview of the model. The output of this whole network is a single number which is the prediction for the binding affinity value for the input drug–target pair. The model is trained to minimize the mean squared error (MSE) between the actual and the predicted binding affinity values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.4}$$

where n is the number of drug-target pairs, Y_i is the the real binding affinity values for the ith input pair and $\hat{Y}_i$ is the predicted value for the same input pair.

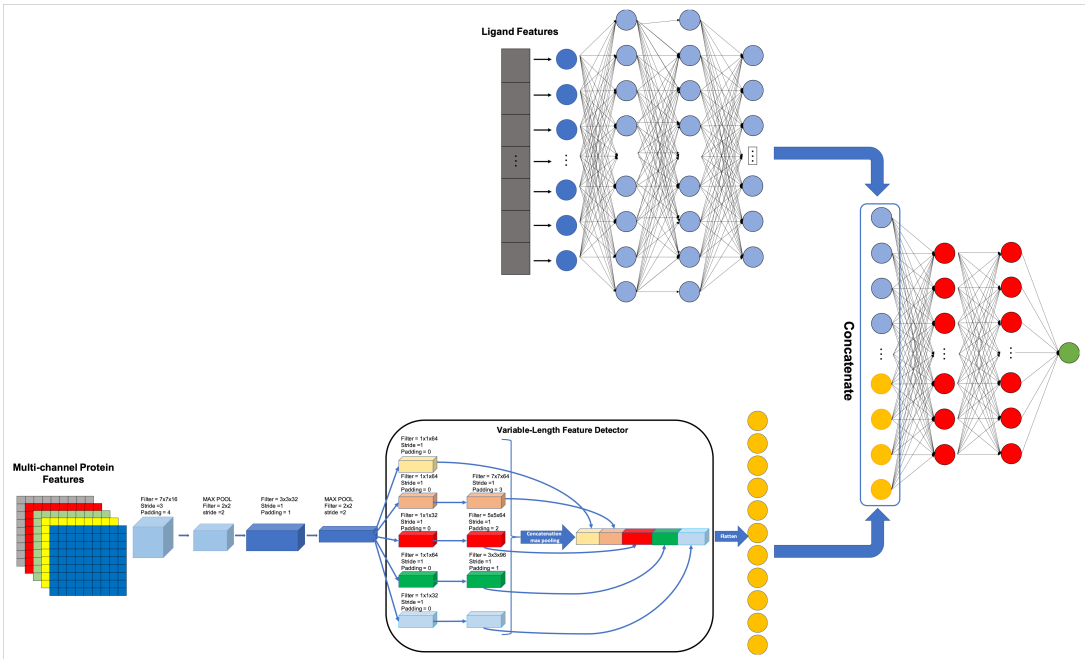

Figure 3.4: Architecture used in the study. Adapted from [30]

3.5 Architecture Used in GO Prediction

The neural network model used in this task is quite similar to the one used in DTI prediction. Since the GO term of a protein has nothing to do with compounds, the corresponding branch is removed from the network. As there are no compound features to concatenate, the flattened output of the inception model is fed to three consecutive fully connected layers. The model is optimized to minimize the cross entropy loss. Cross entropy loss is defined as:

$$CE = -\sum_{i=1}^{C} Y_i log(\hat{Y}_i) \tag{3.5}$$

Where Y_i is the ground truth, $\hat{Y}_i$ is the output of the model and C is the number of classes (GO terms).

CHAPTER 4

DRUG-TARGET INTERACTION PREDICTION

4.1 Introduction

Drug-target interaction is the binding of a drug, a chemical substance, to a target which in turn results in a change in the function of the target [41]. A drug is more specifically defined in pharmaceutical terms as "a chemical substance, typically of known structure, which, when administered to a living organism, produces a biological effect" [42]. Target proteins also referred to as biological targets, are proteins and nucleic acids inside a living organism bound by a drug.

There are thousands of possible targets in the human body. On the other hand, millions of different compounds are available in nature. Thus, it is virtually impossible to exhaustively test every compound-target pair *in-vitro*. To this end, the field of Virtual Screening (VS) was born. Virtual screening is a computational technique that is used for filtering the possible drug candidates down to a feasible number that can be synthesized and tested in wet labs. Virtual screening methods can be divided into following three categories [41, 43, 44].

The first category consists of molecular docking approaches that used 3D structures of compounds and proteins and run simulations in order to decide whether they would bind or not [45, 46, 47, 48]. This approach is limited by the availability of 3D structures of proteins. However, recent advancements that inspired this study will possibly have a positive impact in this field as well.

The second category is ligand-based methods. Ligand-based methods use only the molecular proprieties of the compounds and do not take any information about the

target into account. They are based on the idea that molecules that bind to the same proteins have similar properties. The biggest drawback of these methods is that their performance suffers greatly when the targets have few or no known drugs [49, 50, 51].

The final category of methods, which also includes the method used in this study, are called proteochemometric or chemogenomic methods. Unlike the ligand-based methods, both the compound and the target features are simultaneously used for modeling. Since proteochemometric models do not suffer from the aforementioned drawbacks of the other two categories, they are more popular [52, 53, 54, 55].

There is a wide variety of techniques when it comes to proteochemometric modeling. Yu et al. used Random Forests and Support Vector Machines to tackle the DTI prediction problem [56]. In their work, compounds were represented by descriptors such as; topological descriptors, constitutional descriptors, topological charge indices, and 2D autocorrelations, whereas proteins were represented using structural and physicochemical descriptors such as; autocorrelation descriptors, amphiphilic pseudo-amino acid composition and many others. DeepDTA is a deep learning model that predicts not only whether a drug-target pair will interact or not but the strength of the interaction (binding affinity) as well [57]. On the compound side, DeepDTA uses Simplified Molecular-Input Line-Entry System (SMILES) strings, whereas the proteins are represented using their amino acid sequences. Similar to our study, it uses fixed length inputs, therefore the sequences that are longer than this length are truncated, whereas shorter sequences are zero-padded. By applying convolution to the protein sequence, DeepDTA extracts local residue patterns. DeepConv-DTI is another deep neural network that predicts binding affinity between drug-target pairs. Similar to DeepDTA, protein sequences are fed to a convolutional neural network. Compounds, on the other hand, are represented by binary fingerprints and fed to a fully connected network. Outputs of both networks are then concatenated and used to predict binding affinities [58]. EEG-DTI builds graph networks using multiple biological data types (drug, protein, disease, side-effect). Then by applying graph convolutional networks, they generate low-dimensional features of both the protein and the drugs. Using these low-dimensional features, EEG-DTI predicts whether the input drug-target pair interacts or not [59].

4.2 Dataset

For drug-target interaction prediction, a reduced version of the popular Davis [60] dataset is used. Davis and colleagues tested the interaction of 72 kinase inhibitors against 442 kinases covering a significant portion of the human catalytic protein kinome. Therefore, the dataset consists of 30,056 bioactivity values (Kd values).

In the Davis dataset, 20,931 out of 30,056 of the bioactivity values are recorded as 10 mM meaning no binding was observed in the primary screen, where ligands are screened against the panel at a single concentration of 10 mM; that is, this value is not the actual bioactivity value, but a placeholder number to indicate that there is no binding at all. In a regression model, using these interactions is likely to lead to misleading performance results, as the model can achieve low mean-error by predicting every value around 10 mM. Therefore, these values are removed from the dataset, and the remaining 9125 binding affinity values are named the Filtered Davis dataset.

Of the remaining 442 kinases, 95 of them have mutations. For this reason, their predicted structures are not readily available in AlphaFold DB. Therefore, bioactivity values involving said kinases are also removed from the dataset. The final dataset consists of 6804 bioactivity values, 347 kinases, and 72 compounds.

As mentioned in chapter 3, protein sequences longer than 500 amino acids needed to be shortened which causes information loss. Table 4.1 shows the distribution of protein sequences by their length. As it is visible in the table, two thirds of the proteins were cropped to a certain degree. Moreover, around 20% of the proteins lost at least half of their sequences.

Table 4.1: Distribution of proteins in the DTI dataset by their length

length ≤ 500	$500 <$ length ≤ 1000	length > 1000	Total
102	176	69	347

4.3 Experiments

4.3.1 Method

Four different models with four different sets of input features were used in order to measure the effect of distance matrices. The first model only used 2D encodings. The second model used 2D Encodings and AAindex features (same combination with [30]). The third model used 2D encodings and distance matrices. Finally, the fourth model used 2D encodings, AAindex features, and distance matrices.

Since each model had different features, they all went through individual hyperparameter optimization. As a result, each model has a different combination of learning rate, number of neurons, mini-batch size, and dropout rate. Since this study measures the impact of protein features, hyperparameters involving the compound branch of the network remained untouched.

After the selection of hyperparameters, each model was run five times in order to account for the stochasticity of the learning process. In order to evaluate the models, a five-fold cross-validation approach was taken. In five-fold cross-validation, dataset is split into 5 equal subsets. In each run of the model, one these subsets is used as a test set and other four are used as a train set. The model is run 5 times and average of these 5 runs are reported as the accuracy of the model. Folds are adapted from [30], with the removal of unavailable proteins mentioned in the previous section.

4.3.2 Evaluation Metrics

The results of this study are evaluated in the context of two similar problems. The first of these problems is predicting the binding affinity value between input drug-target pairs. Binding affinity values are continuous; thus, suitable metrics must be chosen. For this problem, results are reported using mean squared error (also the objective function of the neural network), concordance index (CI), and the Spearman rank correlation.

Concordance Index [61] measures whether the predicted binding affinity values of

drug-target pairs are in the same order as the actual values. Such a metric can come in handy when finding the optimal model that could find the best possible drug-target pair, regardless of the correctness of the predicted binding affinity value. It is defined as:

$$CI = \frac{1}{N} \sum_{y_i > y_j} h(\hat{y}_i - \hat{y}_j) \tag{4.1}$$

where N is the number of pairs. y_i and y_j represent the actual affinity values, whereas $\hat{y}_i$ and $\hat{y}_i$ represent the predicted affinity values. $h(x)$ is a step function defined as:

$$h(x) = \begin{cases} 1.0 & \text{if} \quad x > 0 \\ 0.5 & \text{if} \quad x = 0 \\ 0 & \text{if} \quad x < 0 \end{cases} \tag{4.2}$$

The Spearman rank correlation r_s is defined as the Pearson correlation coefficient between the rank variables [62] and computed as follows: For a sample of size n, the n raw scores X_i, Y_i are converted to ranks $\mathrm{R}(X_i), \mathrm{R}(Y_i)$. Then covariance of these rank variables $\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))$ is divided by the standard deviations of the rank variables $\sigma_{\mathrm{R}(X)}$ and $\sigma_{\mathrm{R}(Y)}$

$$r_s = \frac{\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)} \sigma_{\mathrm{R}(Y)}} \tag{4.3}$$

The second problem is modeled as a binary classification where binding/active drug-target pairs are considered positive, whereas non-binding/inactive pairs are considered negatives. However, there is no universal binding affinity value that can separate active and inactive pairs. Therefore results are recorded under three different bioactivity threshold values:1 μM, 100 nM, and 30 nM. In order to evaluate these binary results, Matthews Correlation Coefficient (MCC) [63] metric is used. MCC is defined as:

$$MCC = \frac{\frac{TP}{N} - S \times P}{\sqrt{PS(1-S)(1-P)}} \tag{4.4}$$

$$P = \frac{TP + FP}{N} \tag{4.5}$$

$$S = \frac{TP + FN}{N} \tag{4.6}$$

$$N = TN + TP + FN + FP \tag{4.7}$$

At any given threshold; TN (true negative) represents inactive pairs that are correctly classified as such, TP (true positive) is the number of active pairs that are classified correctly, FN (false negative) is the active pairs that are predicted as inactive, and finally FP (false positive) is the number of inactive pairs that are classified as active pairs.

4.3.3 Results

Table 4.2 shows the results of the model trained with 2D encodings only. Its purpose is to serve as a baseline for future tests. Since the dataset adapted from MDeePred [30] needed to be filtered due to data availability (as explained in the previous section), it would not be fair to use the results reported in the paper. Thus, a model that uses MDeePred features (2D encodings + AAindex matrices) is trained as well. Table 4.3 shows the results of this model. Then, in order to measure the effect of distance matrices, a model is trained with the combination of 2D encodings and distance matrices. Results are shown in table 4.4. Finally, to achieve the best possible result, a model is trained using all features; 2D encodings, AAindex features, and distance matrices. Results are shown in table 4.5. In each of the aforementioned tables, the first five rows are individual runs, whereas the last row is the average value of these individual runs. The average results of each model are also shown in table 4.6 for easier comparison.

The model trained with 2D encodings performs the worst. This is expected because 2D encodings' sole purpose is to distinguish different amino acid pairs from each other, and it does not contain any physical/chemical information. In the second model, with the addition AAindex features, performance is improved. It has a considerable increase in Concordance index and Spearman correlation and a decrease

in MSE. This, again, is expected because by using substitution matrices, the model is fed with more information. The third model, which uses 2D encodings and distance matrices, performed even better than the previous one. Compared to the second model's improvement over the baseline first model, the third model's improvement is almost twofold. Finally, the fourth model, where every available feature matrix is combined, again shows improvement in every performance metric.

Table 4.2: Results of the model trained with 2D encodings.The first five rows shows the individual runs and the last row is the average of these five individual runs.

Concordance Index	Mean Squared Error	Spearman Correlation	MCC 1μM	MCC 100nM	MCC 30nM
0.691	0.622	0.527	0.337	0.465	0.430
0.684	0.623	0.526	0.330	0.457	0.421
0.688	0.623	0.530	0.337	0.452	0.429
0.688	0.627	0.520	0.329	0.453	0.453
0.690	0.613	0.528	0.321	0.473	0.450
0.688	**0.622**	**0.526**	**0.331**	**0.460**	**0.437**

Table 4.3: Results of the model trained with MDeePred features. The first five rows shows the individual runs and the last row is the average of these five individual runs.

Concordance Index	Mean Squared Error	Spearman Correlation	MCC 1μM	MCC 100nM	MCC 30nM
0.695	0.604	0.540	0.335	0.454	0.452
0.687	0.619	0.531	0.334	0.454	0.434
0.691	0.607	0.537	0.332	0.444	0.465
0.699	0.608	0.536	0.338	0.455	0.436
0.687	0.609	0.529	0.326	0.460	0.467
0.692	**0.609**	**0.534**	**0.333**	**0.453**	**0.451**

4.4 Discussion

The results obtained in this part are very much in line with the hypothesis that gave birth to this study. The addition of AAindex features over the 2D encodings resulted in an improvement that is similar to MDeePred, which validates our experimental setup. Then the potency of distance matrices is shown by comparison to AAindex features. Finally, combining all features, a top-performing model is created, which

Table 4.4: Results of the model trained with 2D encodings and distance matrices. The first five rows shows the individual runs and the last row is the average of these five individual runs.

Concordance Index	Mean Squared Error	Spearman Correlation	MCC 1μM	MCC 100nM	MCC 30nM
0.704	0.567	0.567	0.353	0.482	0.456
0.699	0.574	0.565	0.368	0.478	0.448
0.687	0.576	0.565	0.388	0.480	0.448
0.705	0.570	0.563	0.360	0.491	0.453
0.704	0.567	0.566	0.359	0.486	0.462
0.699	**0.571**	**0.565**	**0.362**	**0.483**	**0.453**

Table 4.5: Results of the model trained with every feature matrix. The first five rows shows the individual runs and the last row is the average of these five individual runs.

Concordance Index	Mean Squared Error	Spearman Correlation	MCC 1μM	MCC 100nM	MCC 30nM
0.707	0.567	0.567	0.374	0.484	0.461
0.703	0.569	0.565	0.362	0.481	0.458
0.699	0.569	0.567	0.358	0.488	0.465
0.706	0.567	0.565	0.368	0.486	0.466
0.709	0.571	0.564	0.369	0.489	0.436
0.705	**0.569**	**0.566**	**0.366**	**0.486**	**0.457**

Table 4.6: Average results obtained from each model.

Feature Set	CI	MSE	Spearman	MCC 1μM	MCC 100nM	MCC 30nM
2D Encodings	0.688	0.622	0.526	0.331	0.460	0.437
MDeePred	0.692	0.609	0.534	0.333	0.453	0.451
2D Encodings + Distance Matrices	0.699	0.571	0.565	0.362	0.483	0.453
All	0.705	0.569	0.566	0.366	0.486	0.457

hopefully performs well when tested in complete benchmark datasets.

Considering the lower and upper bounds of the evaluation metrics, the improvements shown in the results might seem insignificant. However, one has to keep in mind that the models used in this part a proteochemometric, meaning that they depend on both the compound and the protein features. For example, even if the models were trained with meaningless protein features, they would still perform at a certain level by memorizing the compounds. This creates a potential lower bound to performance. Moreover, even if we had the theoretically best possible protein input features and the best possible feature extractor on the protein side, the performance of the overall model would be restricted by the compound side, and that creates a hypothetical upper pound to performance. Said lower bound was not studied in this scope, and this hypothetical upper bound is theoretically impossible to find. Thus, little improvements may not be so little in perspective of the aforementioned bounds.

CHAPTER 5

GENE ONTOLOGY PREDICTION

5.1 Introduction

In a living organism, proteins have many different roles. Some proteins are enzymes; they facilitate biochemical reactions. Antibodies are also proteins; they identify and neutralize foreign substances and fight infections. Some proteins are called *contractile proteins*; they make muscles contract. *Insulin* controls our blood sugar levels, whereas *hemoglobin* transports oxygen in our body. It is obvious that proteins' importance to our bodies is indisputable. Knowing a protein's function can greatly help scientists with their efforts in curing related diseases. For example, if the insulin's function was not known, diabetes would not be cured.

Despite its importance, we actually know the function of only a small number of proteins. The number of proteins manually annotated by experts roughly accounts for the 0.5% of the proteins available in UniProt [64]. This stems from the fact that, just as it was in drug-target interaction case, experimentally determining a protein's function is costly. This made scientist resort to *in-silico* methods.

The Gene Ontology (GO) is an initiative to create consistent descriptions of gene and gene product attributes across all species [65]. Gene Ontology covers proteins in three different domains;

- Molecular Function (MF) describes activities that happen at the molecular level, such as *catalysis* or *transport*.
- Biological Process (BP) describes larger processes, or 'biological programs' accomplished by multiple molecular activities, such as *DNA repair* or *signal*

transduction.

- Cellular Component (CC) represents parts of a cell or its extracellular environment, such as *mitochondrion* or *ribosome*.

GO terms create a directed acyclic graph (referred to as the GO hierarchy) according to their hierarchical relationships. In a GO hierarchy graph, a child term (node) is more specialized than its parent term. For example, in Figure 5.1, hexose biosynthetic process (GO:0019319) is a subtype of every other GO term in the graph. This means that a protein with GO term GO:0019319 can also be classified as any of the parent terms.

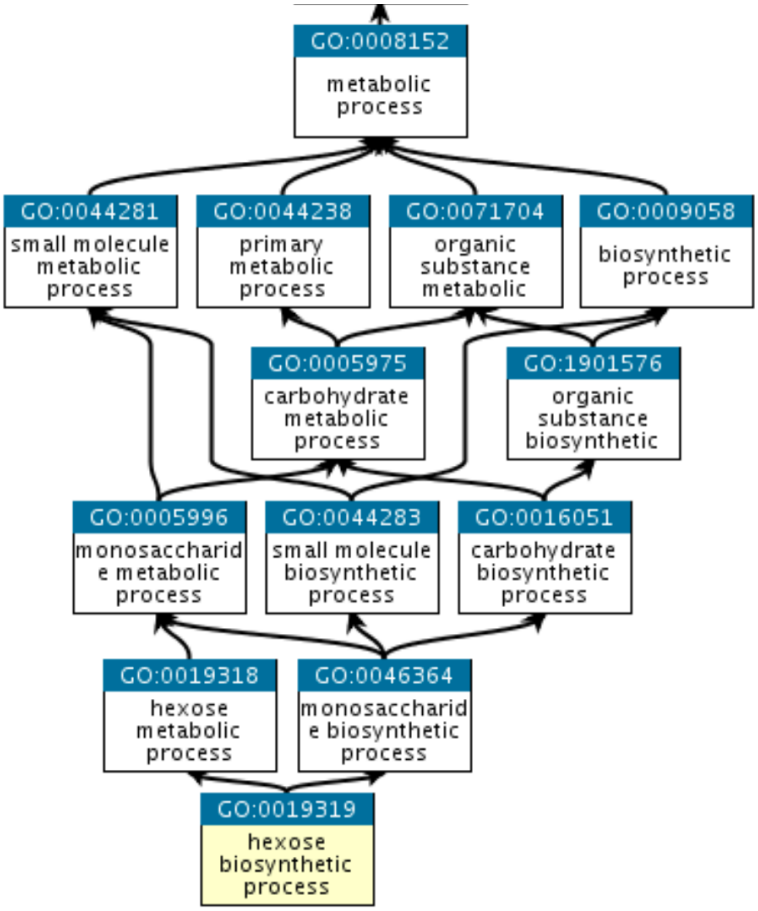

Figure 5.1: Example GO hierarchy. Adapted from http://geneontology.org/docs/ontology-documentation/

Computational protein function prediction received a lot of traction from researchers [66, 67, 68]. The Critical Assessment of protein Function Annotation algorithms (CAFA)

is a challenge that aims to evaluate new computational methods that are capable of predicting Gene Ontology (GO) terms for proteins based on their sequences [69]. The 4th and latest edition of CAFA received 126 different submissions from 56 different research groups.

MS-kNN was one of the most successful methods in the 2nd CAFA challenge. MS-kNN works by finding k-nearest neighbors of the query protein with regards to different similarity metrics such as sequence similarity. Later, by applying weighted averaging to these neighbors' functions, predictions are made for the query protein [70]. GOLabeler was the top performing method in the 3rd CAFA challenge. GOLabeler incorporates several different classifiers trained with different protein features. Candidate predictions from said models are later fed into a learning to rank (LTR) model, which then returns the final GO term prediction [71]. DEEPred is a multi-task, feed-forward deep neural network that uses SPMap [72] features to represent proteins. Being a multi-task network, DEEPred is trained to predict multiple GO terms corresponding to different levels in the GO hierarchy, at the same time [73].

5.2 Dataset

The dataset used in this part was adopted from Protein RepresentatiOn BEnchmark (PROBE) [64] and filtered according to the requirement and limitations of the proposed method. The authors have acquired their data from the "2019_10" version of UniProtKB/Swiss-Prot and UniProtGOA databases. They filtered the dataset so that there were no two protein sequences with 50% similarity. The remaining terms were grouped as "low", "middle", or "high" according to the number of annotated proteins. "Low" groups have 2 to 30 proteins, "middle" groups have 100 to 500, and finally, "high" groups have more than 1000 annotated proteins. Furthermore, GO terms were also grouped according to their position in the GO hierarchy. In the GO hierarchy graph, terms in the first third of the max depth of that branch were considered "shallow", whereas the second and third are considered "normal" and "specific" respectively. Finally, 25 different groups with varying numbers of annotations were obtained, as the two groups had no corresponding GO terms (MF-high-specific and CC-high-specific).

This adopted dataset underwent the following modifications in order to fit the requirements and the limitations of the proposed method:

- "Low" datasets are discarded, as the number of proteins in each was too low to train a deep learning model.
- Protein sequences with more than 1400 amino acids are removed from the datasets. This is because, as it is explained previously, feature matrices are cropped to a certain length. There are two concerns when deciding on this threshold. First, as the protein length increases, cropped versions used in the study get less and less descriptive of the said protein. On the other hand, a lower threshold means fewer proteins to train the model. Analyzing the numbers, 1400 seemed to be the sweet spot. However, the effect of this number is not studied in the scope of this work.
- Proteins with multiple annotations are also removed from the datasets.

Both the number of proteins that are used in the original study and this study can be seen in Table 5.1, Table 5.2, and Table 5.3. Moreover, the class distribution of each dataset can be found in their respective tables. Table 5.4 for Molecular Function; Table 5.5 for Biological Process; Table 5.6 for Cellular Component respectively.

As mentioned earlier, protein sequences longer than 500 amino acids need to be shortened, which causes information loss. In GO term prediction, models were also trained with 250×250 shaped matrices to measure the effect of this number. Table 5.7 shows the distribution of protein sequences used in GO term prediction by their length. According to the table, 47% of the proteins needed to be cropped to a certain extent, whereas 14% lost at least half of their sequences.

5.3 Experiments

5.3.1 Method

For this part, five different model was trained for each dataset. In addition to the ones used in DTI prediction ((i) 2D encodings, (ii) 2D encodings and distance matrices,

Table 5.1: Number of proteins in Molecular Function datasets, before and after modifications

Dataset	Before	After
MF High Shallow	3039	3037
MF High Normal	1324	1323
MF Middle Shallow	613	613
MF Middle Normal	369	369
MF Middle Specific	204	204

Table 5.2: Number of proteins in Biological Process datasets, before and after modifications

Dataset	Before	After
BP High Shallow	3386	3384
BP High Normal	3288	3284
BP High Specific	2025	2022
BP Middle Shallow	896	896
BP Middle Normal	452	451
BP Middle Specific	623	622

Table 5.3: Number of proteins in Cellular Component datasets, before and after modifications

Dataset	Before	After
CC High Shallow	7186	7178
CC High Normal	6478	4602
CC Middle Shallow	413	413
CC Middle Normal	562	531
CC Middle Specific	413	413

Table 5.4: Class distributions for Molecular Function datasets. 'Annotations' column shows the percentage of proteins annotated to each GO term in the dataset. 'Total' column shows the number of the proteins in the dataset.

Dataset	Annotations					Total
MF High Shallow	GO:0046872	GO:0000981	GO:0022890	GO:0004672	GO:0016818	3037
	0.27	0.14	0.28	0.16	0.15	
MF High Normal	GO:0046872			GO:0000981		1323
	0.57			0.43		
MF Middle Shallow	GO:0022835	GO:0005524	GO:0005244	GO:0046943	GO:0036459	513
	0.19	0.21	0.08	0.12	0.40	
MF Middle Normal	GO:0004843	GO:0004714	GO:0003774	GO:0008227	GO:0004866	369
	0.22	0.17	0.19	0.31	0.11	
MF Middle Specific	GO:0003777		GO:0004386	GO:0061733		204
	0.27		0.43	0.30		

Table 5.5: Class distributions for Biological Process datasets. 'Annotations' column shows the percentage of proteins annotated to each GO term in the dataset. 'Total' column shows the number of the proteins in the dataset.

Dataset	Annotations					Total
BP High Shallow	GO:0006886	GO:0051707	GO:0007399	GO:0006259	GO:0007167	3384
	0.19	0.36	0.13	0.10	0.22	
BP High Normal	GO:0048699	GO:0015833	GO:0019752	GO:0070647	GO:0016071	3284
	0.18	0.27	0.26	0.18	0.11	
BP High Specific	GO:0045944		GO:1903507	GO:0001934		2022
	0.43		0.27	0.30		
BP Middle Shallow	GO:0043488	GO:0043312	GO:0051091	GO:0006637	GO:0038096	896
	0.47	0.22	0.17	0.08	0.06	
BP Middle Normal	GO:0051092	GO:0016573	GO:0031146	GO:0071427	GO:0006613	451
	0.29	0.21	0.18	0.16	0.16	
BP Middle Specific	GO:0000209	GO:0071805	GO:1903169	GO:0050773	GO:0031124	622
	0.42	0.13	0.19	0.14	0.12	

Table 5.6: Class distributions for Cellular Component datasets. 'Annotations' column shows the percentage of proteins annotated to each GO term in the dataset. 'Total' column shows the number of the proteins in the dataset.

Dataset	Annotations					Total
CC High Shallow	GO:1990234	GO:1903561	GO:0005740	GO:0070013	GO:000578	7178
	0.52	0.07	0.26	0.06	0.09	
CC High Normal	GO:0031981	GO:0070062	GO:0015630	GO:0005768		4602
	0.53	0.27	0.11	0.09		
CC Middle Shallow	GO:0044853	GO:0036464	GO:0005762	GO:0005747	GO:0008076	413
	0.11	0.18	0.44	0.18	0.09	
CC Middle Normal	GO:0022627	GO:0000502	GO:0034705	GO:0030665	GO:0005925	531
	0.54	0.15	0.06	0.16	0.09	
CC Middle Specific	GO:0016591	GO:0008021	GO:0101002	GO:0005766		413
	0.31	0.28	0.26	0.15		

Table 5.7: Distribution of proteins in the GO dataset by their length

length ≤ 250	$250 <$ length ≤ 500	$500 <$ length ≤ 750	$750 <$ length ≤ 1000	length > 1000	Total
1869	3945	2397	1148	1514	10883

(iii) MDeePred features -2D encodings and AAindex features-, (iv) All available features -2D encodings, AAindex features, and distance matrices-), a model using only the distance matrices is trained as well. With the number of combinations between datasets and set of input features, it was not feasible to make thorough hyperparameter optimization for each model. Hyperparameters are selected after a coarse grid search. Each model was run three times. Models are evaluated with five-fold cross-validation. Since the datasets are very much imbalanced, folds are generated in a stratified manner. This way, all folds contain the same distribution of classes. Moreover, mini-batches used in the training process are sampled using *Weighted Random Sampler* so that each batch has a class distribution similar to the whole dataset.

5.3.2 Evaluation Metric

Since the datasets used in this task are heavily imbalanced, it is generally a better idea to use an evaluation metric other than the simpler ones, such as accuracy. Moreover, this study includes a comparison with PROBE [64], where the authors also used F_1 score to report their results. The F_1 score is the harmonic mean of the precision and recall.

$$F_1 = 2\,\frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{5.1}$$

For the multiclass cases, F_1 score is calculated for each class, then these scores are weighted averaged by the number of true instances for each label which leads to a single F_1 score.

5.3.3 Results

Table 5.8, table 5.9, table 5.10 show the results of Molecular Function, Biological Process, Cellular Component datasets respectively. The results are in line with the ones from the DTI prediction part. Models trained with 2D encodings only performed worst significantly in every dataset. The addition of AAindex features (referred to as 'MDeePred' in the tables) more or less increased the performance in every dataset.

Continuing the pattern from DTI prediction, the combination of 2D encodings and distance matrices significantly outperforms the combination of 2D encodings and AAindex features (MDeePred) in each dataset. Models trained with only the distance matrices produced some thought-provoking results. They outperformed models trained with MDeePred features, cementing the significance of distance matrices. Moreover, they outperformed 2D encodings in ten of the sixteen datasets, where margins were minimal for both sides. At first glance, this sounds unintuitive as it makes 2D encodings seem redundant. Possible reasons for this phenomenon are explored in the next section. Finally, as expected, models which used every available feature achieved the highest F_1 scores.

Table 5.8: F_1 scores achieved on Molecular Function datasets. '2D Encodings' refers to the model trained using only the 2D Encodings, 'Distance Only' refers to the model trained using only the distance matrices, 'MDeePred' refers to the model trained using the features in MDeePred -2D encodings and AAindex features-, 'All' refers to the model trained using 2D Encodings + distance matrices + AAindex features.

Dataset	2D Encoding	Distance Only	MDeePred	2D Encoding + Distance	All
MF High Shallow	0.373	0.630	0.524	0.643	0.680
MF High Normal	0.669	0.847	0.761	0.861	0.867
MF Mid Shallow	0.537	0.712	0.633	0.702	0.742
MF Mid Normal	0.562	0.706	0.629	0.655	0.703
MF Mid Specific	0.545	0.650	0.577	0.643	0.652

5.3.4 Effect of Feature Size

In order to measure the impact of protein sequence length selection, a new model is trained for each dataset using all available features (2D encodings, AAindex features, and distance matrices), this time cropped to 250×250. When cut to 250 amino acids, 83% of the proteins in the dataset faces information loss. Table 5.11 shows the comparison between models trained with 250×250 shaped and 500×500 shaped features. 750×750 matrices were also considered for this part; however, the resulting input did not fit into the memory of the GPU which is used to run the experiments.

Table 5.9: F_1 scores achieved on Biological Process datasets. '2D Encodings' refers to the model trained using only the 2D Encodings, 'Distance Only' refers to the model trained using only the distance matrices, 'MDeePred' refers to the model trained using the features in MDeePred -2D encodings and AAindex features-, 'All' refers to the model trained using 2D Encodings + distance matrices + AAindex features.

Dataset	2D Encoding	Distance Only	MDeePred	2D Encoding + Distance	All
BP High Shallow	0.331	0.375	0.336	0.356	0.375
BP High Normal	0.297	0.425	0.322	0.419	0.430
BP High Specific	0.429	0.479	0.458	0.484	0.496
BP Mid Shallow	0.410	0.482	0.431	0.483	0.503
BP Mid Normal	0.405	0.433	0.409	0.449	0.475
BP Mid Specific	0.372	0.478	0.412	0.457	0.487

Table 5.10: F_1 scores achieved on Cellular Component datasets. '2D Encodings' refers to the model trained using only the 2D Encodings, 'Distance Only' refers to the model trained using only the distance matrices, 'MDeePred' refers to the model trained using the features in MDeePred -2D encodings and AAindex features-, 'All' refers to the model trained using 2D Encodings + distance matrices + AAindex features.

Dataset	2D Encoding	Distance Only	MDeePred	2D Encoding + Distance	All
CC High Shallow	0.430	0.517	0.484	0.505	0.529
CC High Normal	0.430	0.537	0.477	0.523	0.539
CC Mid Shallow	0.415	0.512	0.428	0.487	0.502
CC Mid Normal	0.484	0.550	0.499	0.557	0.579
CC Mid Specific	0.362	0.457	0.380	0.427	0.437

Table 5.11: Comparison of F_1 scores achieved using features with different shapes.

Dataset	500×500	250×250
MF High Shallow	0.680	0.562
MF High Normal	0.867	0.818
MF Mid Shallow	0.742	0.617
MF Mid Normal	0.703	0.598
MF Mid Specific	0.652	0.515
BP High Shallow	0.475	0.364
BP High Normal	0.430	0.362
BP High Specific	0.496	0.484
BP Mid Shallow	0.503	0.465
BP Mid Normal	0.475	0.412
BP Mid Specific	0.487	0.445
CC High Shallow	0.529	0.509
CC High Normal	0.539	0.527
CC Mid Specific	0.502	0.426
CC Mid Normal	0.579	0.473
CC Mid Specific	0.437	0.419

5.3.5 Comparisons with PROBE

In PROBE [64], authors compared several feature embeddings. Those embeddings are; APAAC [19], k-sep biagrams [17], LearnedEmbeddingVec [74], SeqVec [75], Mut2Vec [76], Gene2Vec [77], TCGA_Embedding [78], ProtVec [79], TAPE-BERT_Avg [80], TAPE-BERT_Pool [80], UniRep [81]. They used a Linear Support Vector Classifier for predictions and used five-fold cross validation to evaluate these models.

For each dataset; worst (to set a baseline), median (to act as a middle-of-the pack classifier), and best scoring models are reported from PROBE and compared to best result achieved from this study. It needs to be noted that, scores taken from PROBE might be a result a different model; meaning there is no single model that performs best in every dataset. Also, while selecting the worst model, models with F_1 score less than 0.1 are disregarded. Table 5.12 shows this comparison.

Looking at the table, our models performs better than the worst model in each dataset. While beating median models in Molecular Function dataset, our models' success declines in Biological Process and Cellular Component datasets. Finally, our models seems to be no match for the best models adopted from PROBE. A possible explanation is given in the next section.

5.4 Discussion

The results of this part are in line with what was expected and what was observed in the drug-target interaction part. Distance matrices offer very significant improvements, outperforming a bunch of substitution matrices. Moreover, it again seems to be the case that a model's performance improves with the addition of input features. However, as mentioned earlier, the addition of 2D encodings does not seem to help with the performance of distance matrices and, in most cases, even degrades them. Like any other deep neural network, our models do not guarantee the most optimal result. Thus, considering how small is the difference between F_1 scores, it is hard to say if distance only models perform better than the ones with 2D encodings because the said difference could be a result of randomness. Under the assumption that both

Table 5.12: Comparison with PROBE results. The 'worst', the 'median' and the 'best' performing models from PROBE are compared to our best performing model.

Dataset	Worst Model	Median Model	Best Model	Our Model
MF High Shallow	0.18	0.58	0.87	0.68
MF High Normal	0.69	0.85	0.94	0.87
MF Mid Shallow	0.52	0.66	0.94	0.74
MF Mid Normal	0.60	0.65	0.92	0.70
MF Mid Specific	0.55	0.64	0.91	0.65
BP High Shallow	0.17	0.41	0.49	0.38
BP High Normal	0.17	0.43	0.67	0.43
BP High Specific	0.41	0.56	0.69	0.50
BP Mid Shallow	0.28	0.51	0.69	0.50
BP Mid Normal	0.24	0.57	0.78	0.48
BP Mid Specific	0.32	0.50	0.75	0.49
CC High Shallow	0.47	0.54	0.72	0.53
CC High Normal	0.41	0.52	0.69	0.54
CC Mid Specific	0.40	0.60	0.77	0.50
CC Mid Normal	0.48	0.68	0.74	0.58
CC Mid Specific	0.16	0.49	0.54	0.44

models perform the same, 2D encodings' inability to improve performance needs explanation. It is possible that 2D encodings are not bringing in any valuable information to the network in the presence of distance information as they do not contain any physical/chemical information regarding the amino acids. Because it is evident from the results that AAindex features containing biological, chemical, and physical information indeed result in improvements.

Our already proven [30] models, with all of their bells and whistles, even with the addition of distance matrices, are unable to compete with relatively simple SVM classifiers reported in PROBE [64]. The most obvious reasoning is that our model's complexity creates a double-edged sword. On average, GO term prediction models used in this study have around 40.000.000 (forty million) trainable parameters. Our most prominent feature set has the shape of [500,500,6]. On the other hand, the average number of observations in high datasets is around 3400, whereas, in the middle datasets, this number reduces to around 550. The assumption here is that such a number of observations is not enough to successfully train a complex model that is used in this study, whereas SVM models, being less complex, did not face the same problem. It is not possible to train an SVM using the data representations of this study, and the features reviewed in PROBE will not work on our architecture. Therefore the only way to prove this claim is to test both methods with a dataset that is curated to include more observations.

CHAPTER 6

CONCLUSION

AlphaFold's success in the latest CASP challenge generated much hype around the scientific communities. This study was an attempt to find out whether it was possible to profit from these predictions in the context of our group's research directions. To this end, we have employed 2D protein distance matrices, a relatively primitive form of actual 3D structures. This made the adaptation of the model used in MDeePred possible. This system combining distance matrices and MDeePred architecture was tested on two problems: drug-target interaction prediction and Gene Ontology term prediction.

In drug-target interaction prediction, the effect of distance matrices was measured by testing different models trained with different sets of features, including distance matrices. With these experiments, distance matrices were found to be more powerful than various generic amino acid substitution matrices, including the very famous BLOSUM62. Plus, it is shown that the addition of distance matrices increases the performance of MDeePred. Nevertheless, since the dataset used in this part can be considered novel, this last claim needs to be tested further.

In Gene Ontology term prediction, similar results were seen. Again, distance matrices looked like the most powerful representation among the set of features that were used in training the models. However, compared to other protein representations reviewed in the PROBE paper, our approach did not seem so successful. A possible explanation related to the size of datasets is given in the corresponding chapter.

In conclusion, 2D distance matrices constructed using AlphaFold structure predictions are shown to be powerful representations of protein sequences. It is possible

that with more specialized neural network architectures, distance matrices could shine even more. Regardless, it is expected that AlphaFold or other protein structure prediction systems will keep gaining traction from bioinformaticians unless a breakthrough happens in X-ray crystallography.

6.1 Future Work

As explained in section 4.2, the Davis dataset used in DTI prediction was trimmed due to limitations in protein structure availability. This made a direct comparison with literature impossible. With the introduction of methods such as OmegaFold [82], structure predictions are becoming easily accessible. The approach proposed in this study needs to be tested with complete datasets and compared to literature.

Authors of MDeePred chose the AAindex substation matrices in the absence of the distance matrices. For example, SIMK990101 and ZHAC000103 matrices were meant to incorporate information about a protein's 3D structure. With the addition of distance matrices, these features might become redundant. It is possible that there exist different substitution matrices that synergize better with distance matrices.

Due to lack of computational resources, feature matrices used in the study are limited to 500×500. However, in section 5.3, it is shown that bigger features lead to better results. With adequate resources, bigger matrices could be tested in the future.

REFERENCES

[1] U. D. J. G. I. H. T. . B. E. . P. P. . R. P. . W. S. . S. T. . D. N. . C. J.-F. . O. A. . L. S. . E. C. . U. E. . F. M. 4, R. G. S. C. S. Y. . F. A. . H. M. . Y. T. . T. A. . I. T. . K. C. . W. H. . T. Y. . T. T. 9, Genoscope, C. U.-. W. J. . H. R. . S. W. . A. F. . B. P. . B. T. . P. E. . R. C. . W. P. 10, I. o. M. B. R. A. . P. M. . N. G. . T. S. . R. A. . Department of Genome Analysis, G. S. C. S. D. R. . D.-S. L. . R. M. . W. K. . L. H. M. . D. J. 11, B. G. I. G. C. Y. H. . Y. J. . W. J. . H. G. . G. J. 15, *et al.*, "Initial sequencing and analysis of the human genome," *nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[2] "Uniprot: the universal protein knowledgebase in 2021," *Nucleic acids research*, vol. 49, no. D1, pp. D480–D489, 2021.

[3] M. C. Thompson, T. O. Yeates, and J. A. Rodriguez, "Advances in methods for atomic resolution macromolecular structure determination," *F1000Research*, vol. 9, 2020.

[4] J. Moult, J. T. Pedersen, R. Judson, and K. Fidelis, "A large-scale experiment to assess protein structure prediction methods," 1995.

[5] E. Callaway, "'it will change everything': Deepmind's ai makes gigantic leap in solving protein structures," Nov 2020.

[6] H. Huang and X. Gong, "A review of protein inter-residue distance prediction," *Current Bioinformatics*, vol. 15, no. 8, pp. 821–830, 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[8] J. Xu, "Distance-based protein folding powered by deep learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 34, pp. 16856–16865, 2019.

[9] W. Ding and H. Gong, "Predicting the real-valued distances between residue pairs for proteins," *arXiv preprint arXiv:1912.06306*, 2019.

[10] T. Wu, Z. Guo, J. Hou, and J. Cheng, "Deepdist: real-value inter-residue distance prediction with deep residual convolutional network," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–17, 2021.

[11] J. Rahman, M. Newton, M. K. B. Islam, and A. Sattar, "Enhancing protein inter-residue real distance prediction by scrutinising deep learning models," *Scientific Reports*, vol. 12, no. 1, pp. 1–13, 2022.

[12] L. Bartoli, E. Capriotti, P. Fariselli, P. L. Martelli, and R. Casadio, "The pros and cons of predicting protein contact maps," *Protein Structure Prediction*, pp. 199–217, 2008.

[13] J. Xie, W. Ding, L. Chen, Q. Guo, and W. Zhang, "Advances in protein contact map prediction based on machine learning," *Medicinal Chemistry*, vol. 11, no. 3, pp. 265–270, 2015.

[14] H. Huang and X. Gong, "A review of protein inter-residue distance prediction," *Current Bioinformatics*, vol. 15, no. 8, pp. 821–830, 2020.

[15] R. Trivedi and H. A. Nagarajaram, "Substitution scoring matrices for proteins-an overview," *Protein Science*, vol. 29, no. 11, pp. 2150–2163, 2020.

[16] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "Aaindex: amino acid index database, progress report 2008," *Nucleic acids research*, vol. 36, no. suppl_1, pp. D202–D205, 2007.

[17] H. Saini, G. Raicar, A. Sharma, S. Lal, A. Dehzangi, R. Ananthanarayanan, J. Lyons, N. Biswas, and K. K. Paliwal, "Protein structural class prediction via k-separated bigrams using position specific scoring matrix," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 18, no. 4, pp. 474–479, 2014.

[18] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246–255, 2001.

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

[20] D. Ofer, N. Brandes, and M. Linial, "The language of proteins: Nlp, machine learning & protein sequences," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 1750–1758, 2021.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[22] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, pp. 1188–1196, PMLR, 2014.

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

[24] M. K. Mejía-Guerra and E. S. Buckler, "A k-mer grammar analysis to uncover maize regulatory architecture," *BMC plant biology*, vol. 19, no. 1, pp. 1–17, 2019.

[25] P. Ng, "dna2vec: Consistent vector representations of variable-length k-mers," *arXiv preprint arXiv:1701.06279*, 2017.

[26] J. Choi, I. Oh, S. Seo, and J. Ahn, "G2vec: Distributed gene representations for identification of cancer prognostic genes," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[27] A. Dutta, T. Dubey, K. K. Singh, and A. Anand, "Splicevec: distributed feature representations for splice junction prediction," *Computational biology and chemistry*, vol. 74, pp. 434–441, 2018.

[28] R. You, X. Huang, and S. Zhu, "Deeptext2go: Improving large-scale protein function prediction with deep semantic text representation," *Methods*, vol. 145, pp. 82–90, 2018.

[29] Y. Liu, Y. Liu, G. Wang, Y. Cheng, S. Bi, and X. Zhu, "Bert-kgly: A bidirectional encoder representations from transformers (bert)-based model for predict-

ing lysine glycation site for homo sapiens," *Frontiers in Bioinformatics*, p. 12, 2022.

[30] A. S. Rifaioglu, R. Cetin Atalay, D. Cansen Kahraman, T. Doğan, M. Martin, and V. Atalay, "Mdeepred: novel multi-channel protein featurization for deep learning-based binding affinity prediction in drug discovery," *Bioinformatics*, vol. 37, no. 5, pp. 693–704, 2021.

[31] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, *et al.*, "Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models," *Nucleic acids research*, vol. 50, no. D1, pp. D439–D444, 2022.

[32] H. K. Ho, M. J. Kuiper, and R. Kotagiri, "Pconpy—a python module for generating 2d protein maps," *Bioinformatics*, vol. 24, no. 24, pp. 2934–2935, 2008.

[33] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff, and D. Baker, "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins," *Proteins: Structure, Function, and Bioinformatics*, vol. 34, no. 1, pp. 82–95, 1999.

[34] C. Zhang and S.-H. Kim, "Environment-dependent residue contact energies for proteins," *Proceedings of the National Academy of Sciences*, vol. 97, no. 6, pp. 2550–2555, 2000.

[35] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," *Proceedings of the National Academy of Sciences*, vol. 89, no. 22, pp. 10915–10919, 1992.

[36] R. Grantham, "Amino acid difference formula to help explain protein evolution," *science*, vol. 185, no. 4154, pp. 862–864, 1974.

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[38] Y.-T. Zhou and R. Chellappa, "Computation of optical flow using a neural network.," in *ICNN*, pp. 71–78, 1988.

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[41] K. Sachdev and M. K. Gupta, "A comprehensive review of computational techniques for the prediction of drug side effects," *Drug Development Research*, vol. 81, no. 6, p. 650.

[42] H. P. Rang and M. M. Dale, *Rang and Dale's Pharmacology*. Elsevier/Churchill Livingstone, 2014.

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

[44] A. Ezzat, M. Wu, X.-L. Li, and C.-K. Kwoh, "Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey," *Briefings in bioinformatics*, vol. 20, no. 4, pp. 1337–1357, 2019.

[45] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui, "Molecular docking: a powerful approach for structure-based drug discovery," *Current computer-aided drug design*, vol. 7, no. 2, pp. 146–157, 2011.

[46] T. Cheng, Q. Li, Z. Zhou, Y. Wang, and S. H. Bryant, "Structure-based virtual screening for drug discovery: a problem-centric review," *The AAPS journal*, vol. 14, no. 1, pp. 133–141, 2012.

[47] E. Lionta, G. Spyrou, D. K Vassilatis, and Z. Cournia, "Structure-based virtual screening for drug discovery: principles, applications and recent advances," *Current topics in medicinal chemistry*, vol. 14, no. 16, pp. 1923–1938, 2014.

[48] N. S. Pagadala, K. Syed, and J. Tuszynski, "Software for molecular docking: a review," *Biophysical reviews*, vol. 9, no. 2, pp. 91–102, 2017.

[49] F. L. Stahura and J. Bajorath, "New methodologies for ligand-based virtual screening," *Current pharmaceutical design*, vol. 11, no. 9, pp. 1189–1202, 2005.

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

[51] I. I. Baskin, "The power of deep learning to ligand-based novel drug discovery," *Expert Opinion on Drug Discovery*, vol. 15, no. 7, pp. 755–764, 2020.

[52] D. Rognan, "Chemogenomic approaches to rational drug design," *British journal of pharmacology*, vol. 152, no. 1, pp. 38–52, 2007.

[53] G. J. Van Westen, J. K. Wegner, A. P. IJzerman, H. W. Van Vlijmen, and A. Bender, "Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets," *MedChemComm*, vol. 2, no. 1, pp. 16–30, 2011.

[54] T. Qiu, J. Qiu, J. Feng, D. Wu, Y. Yang, K. Tang, Z. Cao, and R. Zhu, "The recent progress in proteochemometric modelling: focusing on target descriptors, cross-term descriptors and application scope," *Briefings in Bioinformatics*, vol. 18, no. 1, pp. 125–136, 2017.

[55] B. J. Bongers, A. P. IJzerman, and G. J. Van Westen, "Proteochemometrics–recent developments in bioactivity and selectivity modeling," *Drug Discovery Today: Technologies*, vol. 32, pp. 89–98, 2019.

[56] H. Yu, J. Chen, X. Xu, Y. Li, H. Zhao, Y. Fang, X. Li, W. Zhou, W. Wang, and Y. Wang, "A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data," *PloS one*, vol. 7, no. 5, p. e37608, 2012.

[57] H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.

[58] I. Lee, J. Keum, and H. Nam, "Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences," *PLoS computational biology*, vol. 15, no. 6, p. e1007129, 2019.

[59] J. Peng, Y. Wang, J. Guan, J. Li, R. Han, J. Hao, Z. Wei, and X. Shang, "An end-to-end heterogeneous graph representation learning-based framework for drug–target interaction prediction," *Briefings in Bioinformatics*, vol. 22, no. 5, p. bbaa430, 2021.

[60] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar, "Comprehensive analysis of kinase inhibitor selectivity," *Nature Biotechnology*, vol. 29, no. 11, p. 1046–1051, 2011.

[61] M. Gönen and G. Heller, "Concordance probability and discriminatory power in proportional hazards regression," *Biometrika*, vol. 92, no. 4, pp. 965–970, 2005.

[62] J. L. Myers and A. Well, *Research design and statistical analysis*. Lawrence Erlbaum Associates, 2003.

[63] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[64] S. Unsal, H. Ataş, M. Albayrak, K. Turhan, A. C. Acar, and T. Doğan, "Evaluation of methods for protein representation learning: A quantitative analysis," *bioRxiv*, 2020.

[65] G. O. Consortium, "The gene ontology (go) database and informatics resource," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D258–D261, 2004.

[66] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur, *et al.*, "A large-scale evaluation of computational protein function prediction," *Nature methods*, vol. 10, no. 3, pp. 221–227, 2013.

[67] Y. Zhao, J. Wang, J. Chen, X. Zhang, M. Guo, and G. Yu, "A literature review of gene function prediction by modeling gene ontology," *Frontiers in Genetics*, vol. 11, p. 400, 2020.

[68] R. Bonetta and G. Valentino, "Machine learning techniques for protein function prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 88, no. 3, pp. 397–413, 2020.

[69] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, M. N. Hamid, *et al.*, "The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome biology*, vol. 20, no. 1, pp. 1–23, 2019.

[70] L. Lan, N. Djuric, Y. Guo, and S. Vucetic, "Ms-k nn: protein function prediction by integrating multiple data sources," in *BMC bioinformatics*, vol. 14, pp. 1–10, BioMed Central, 2013.

[71] R. You, Z. Zhang, Y. Xiong, F. Sun, H. Mamitsuka, and S. Zhu, "Golabeler: improving sequence-based large-scale protein function prediction by learning to rank," *Bioinformatics*, vol. 34, no. 14, pp. 2465–2473, 2018.

[72] O. S. Sarac, Ö. Gürsoy-Yüzügüllü, R. Cetin-Atalay, and V. Atalay, "Subsequence-based feature map for protein function classification," *Computational biology and chemistry*, vol. 32, no. 2, pp. 122–130, 2008.

[73] A. Sureyya Rifaioglu, T. Doğan, M. Jesus Martin, R. Cetin-Atalay, and V. Atalay, "Deepred: automated protein function prediction with multi-task feed-forward deep neural networks," *Scientific reports*, vol. 9, no. 1, pp. 1–16, 2019.

[74] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold, "Learned protein embeddings for machine learning," *Bioinformatics*, vol. 34, no. 15, pp. 2642–2648, 2018.

[75] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nachaev, F. Matthes, and B. Rost, "Modeling the language of life-deep learning protein sequences. biorxiv," 2019.

[76] S. Kim, H. Lee, K. Kim, and J. Kang, "Mut2vec: distributed representation of cancerous mutations," *BMC medical genomics*, vol. 11, no. 2, pp. 57–69, 2018.

[77] J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, and D. Zhi, "Gene2vec: distributed representation of genes based on co-expression," *BMC genomics*, vol. 20, no. 1, pp. 7–15, 2019.

[78] C. T. Choy, C. H. Wong, and S. L. Chan, "Infer related genes from large scale gene expression dataset with embedding," *bioRxiv*, p. 362848, 2018.

[79] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, p. e0141287, 2015.

[80] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, and Y. Song, "Evaluating protein transfer learning with tape," *Advances in neural information processing systems*, vol. 32, 2019.

[81] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, "Unified rational protein engineering with sequence-based deep representation learning," *Nature methods*, vol. 16, no. 12, pp. 1315–1322, 2019.

[82] R. Wu, F. Ding, R. Wang, R. Shen, X. Zhang, S. Luo, C. Su, Z. Wu, Q. Xie, B. Berger, J. Ma, and J. Peng, "High-resolution de novo structure prediction from primary sequence," *bioRxiv*, 2022.

Appendix A

HYPERPARAMETERS OF DTI PREDICTION MODELS

Table A.1: Hyperparameters of the models used in drug-target interaction prediction.

Dataset	Mini-batch Size	Number of Epochs	Learning Rate	Number of Neurons in FC Layers
2D Encodings	32	200	0.00001	[1024,1024]
MDeePred features	32	200	0.00001	[1024,1024]
2D Encodings + Distance	32	200	0.00005	[1024,512]
All features	32	300	0.0005	[2048,1024]

Appendix B

PARAMETERS OF GO PREDICTION MODELS

Table B.1: Hyperparameters of the models trained using only the 2D encodings.

Dataset	Mini-batch Size	Number of Epochs	Learning Rate	Number of Neurons in FC Layers
MF High Datasets	16	18	0.00001	[1024,8192,512]
MF Mid Datasets	16	12	0.00005	[1024,8192,512]
BP High Datasets	16	18	0.00001	[1024,8192,512]
BP Mid Datasets	16	15	0.00001	[1024,4096,64]
CC High Datasets	16	18	0.00001	[1024,4096,64]
CC Mid Datasets	16	15	0.00001	[1024,4096,64]

Table B.2: Hyperparameters of the models trained using only the distance matrices.

Dataset	Mini-batch Size	Number of Epochs	Learning Rate	Number of Neurons in FC Layers
MF High Datasets	16	12	0.0001	[1024,8192,512]
MF Mid Datasets	16	10	0.0001	[1024,4096,128]
BP High Datasets	16	12	0.00001	[1024,8192,512]
BP Mid Datasets	16	10	0.00001	[1024,4096,64]
CC High Datasets	16	12	0.0001	[1024,4096,128]
CC Mid Datasets	16	10	0.00001	[1024,8192,512]

Table B.3: Hyperparameters of the models trained using the feature combination used in MDeePred.

Dataset	Mini-batch Size	Number of Epochs	Learning Rate	Number of Neurons in FC Layers
MF High Datasets	16	25	0.0001	[1024,8192,512]
MF Mid Datasets	16	20	0.00001	[1024,8192,512]
BP High Datasets	16	25	0.00001	[1024,4096,128]
BP Mid Datasets	16	20	0.00001	[1024,4096,128]
CC High Datasets	16	25	0.0001	[1024,4096,128]
CC Mid Datasets	16	20	0.00001	[1024,8192,512]

Table B.4: Hyperparameters of the models trained using the combination of 2D encodings and distance matrices.

Dataset	Mini-batch Size	Number of Epochs	Learning Rate	Number of Neurons in FC Layers
MF High Datasets	16	22	0.0001	[1024,4096,128]
MF Mid Datasets	16	20	0.0001	[1024,4096,128]
BP High Datasets	16	22	0.00001	[1024,4096,128]
BP Mid Datasets	16	20	0.00005	[1024,4096,128]
CC High Datasets	16	22	0.00005	[1024,8192,512]
CC Mid Datasets	16	20	0.00001	[1024,4096,128]

Table B.5: Hyperparameters of the models trained using all features.

Dataset	Mini-batch Size	Number of Epochs	Learning Rate	Number of Neurons in FC Layers
MF High Datasets	8	32	0.0001	[1024,4096,128]
MF Mid Datasets	16	25	0.00005	[1024,4096,64]
BP High Datasets	8	32	0.00005	[1024,4096,128]
BP Mid Datasets	8	22	0.0001	[1024,8192,512]
CC High Datasets	16	30	0.0001	[1024,4096,128]
CC Mid Datasets	8	25	0.00001	[1024,2048,512]

Appendix C

GO TERMS

Table C.1: GO terms of the Molecular Function datasets.

ID	Term	ID	Term
GO:0046872	metal ion binding	GO:0000981	DNA-binding transcription factor activity, RNA polymerase II-specific
GO:0022890	inorganic cation transmembrane transporter activity	GO:0004672	protein kinase activity
GO:0016818	hydrolase activity, acting on acid anhydrides, in phosphorus-containing anhydrides	GO:0046872	metal ion binding
GO:0000981	DNA-binding transcription factor activity, RNA polymerase II-specific	GO:0022835	transmitter-gated channel activity
GO:0005524	ATP binding	GO:0005244	voltage-gated ion channel activity
GO:0046943	carboxylic acid transmembrane transporter activity	GO:0036459	thiol-dependent ubiquitinyl hydrolase activity
GO:0004843	cysteine-type deubiquitinase activity	GO:0004714	transmembrane receptor protein tyrosine kinase activity
GO:0003774	cytoskeletal motor activity	GO:0008227	G protein-coupled amine receptor activity
GO:0004866	endopeptidase inhibitor activit	GO:0003777	microtubule motor activity
GO:0004386	helicase activity	GO:0061733	peptide-lysine-N-acetyltransferase activity

Table C.2: GO terms of the Biological Process datasets.

ID	Term	ID	Term
GO:0031124	mRNA 3'-end processing	GO:0006886	intracellular protein transport
GO:0051707	response to other organism	GO:0007399	nervous system development
GO:0007167	enzyme-linked receptor protein signaling pathway	GO:0006259	DNA metabolic process
GO:0048699	generation of neurons	GO:0015833	peptide transport
GO:0019752	carboxylic acid metabolic process	GO:0070647	protein modification by small protein conjugation or removal
GO:0050773	regulation of dendrite development	GO:0016071	mRNA metabolic process
GO:0045944	positive regulation of transcription by RNA polymerase II	GO:1903507	negative regulation of nucleic acid-templated transcription
GO:0001934	positive regulation of protein phosphorylation	GO:0043488	regulation of mRNA stability
GO:0043312	neutrophil degranulation	GO:0051091	positive regulation of DNA-binding transcription factor activity
GO:0006637	acyl-CoA metabolic process	GO:0038096	Fc-gamma receptor signaling pathway involved in phagocytosis 79 annotations
GO:0051092	positive regulation of NF-kappaB transcription factor activity	GO:0016573	histone acetylation
GO:0031146	SCF-dependent proteasomal ubiquitin-dependent protein catabolic process	GO:0071427	obsolete mRNA-containing ribonucleoprotein complex export from nucleus
GO:0006613	cotranslational protein targeting to membrane	GO:0000209	protein polyubiquitination
GO:0071805	potassium ion transmembrane transport	GO:1903169	regulation of calcium ion transmembrane transport

Table C.3: GO terms of the Cellular Component datasets.

ID	Term	ID	Term
GO:1990234	transferase complex	GO:1903561	extracellular vesicle
GO:0005740	mitochondrial envelope	GO:0070013	intracellular organelle lumen
GO:000578	GINS complex	GO:0031981	nuclear lumen
GO:0070062	extracellular exosome	GO:0015630	microtubule cytoskeleton
GO:000576	GINS complex	GO:0044853	plasma membrane raft
GO:0036464	cytoplasmic ribonucleoprotein granule	GO:0005762	mitochondrial large ribosomal subunit
GO:0005747	mitochondrial respiratory chain complex I	GO:0008076	voltage-gated potassium channel complex
GO:0022627	cytosolic small ribosomal subunit	GO:0000502	proteasome complex
GO:0034705	potassium channel complex	GO:0030665	clathrin-coated vesicle membrane
GO:0005925	focal adhesion	GO:0016591	RNA polymerase II, holoenzyme
GO:0008021	synaptic vesicle	GO:0101002	ficolin-1-rich granule
GO:0005766	primary lysosome		