Machine Learning and Deep Learning Techniques for CPI & Binding Affinity

Roziena Badree and Mengriu Mao

Spring 2023

Contents

1 Introduction **2**

2 Proteins **3**

3 Approach **5**

3.1 Datasets 5

3.2 BACPI Model 6

3.2.1 Compound Representation 6

3.2.2 Protein Representation 8

3.2.3 CPI & Binding Affinity Prediction 9

3.3 Evolutionary Scale Model (ESM) 12

3.3.1 Compound Representation 12

3.3.2 Protein Representation 13

3.3.3 CPI & Binding Affinity Prediction 13

3.4 Models & Evaluation 13

4 Results & Discussion **15**

5 Conclusion **17**

6 Appendix A **20**

7 Appendix B **21**

Introduction

The identification of compound–protein interactions (CPIs) is a crucial step in drug discovery. Since the experimental determination of CPIs is both expensive and time-consuming, the computational model provides a promising and efficient alternative [3].

CPI prediction is a binary classification task, which determines if there is a molecular interaction between a drug (we refer to a drug as a compound in this project) and protein target while compound–protein binding affinity prediction is a regression task, which focuses on predicting a continuous value named binding affinity that represents the binding strength between a compound and the target protein [3, 5].

We found two promising techniques that appear to outperform others when predicting CPI and binding affinity: a bi-directional attention neural network named BACPI and Evolutionary Scale Modeling (ESM). Li et al. in their recent experiment "BACPI: a bi-directional attention neural network for compound–protein interaction and binding affinity prediction" created an end-to-end neural network model to predict CPI and binding affinity [3]. Google − through its AlphaFold project − and Meta are currently building language models trained on the amino acid sequences of millions of diverse proteins to predict how a protein will fold thereby learning patterns about the underlying structure in the sequences. A protein's folding is essential to understanding its structure and function, which can be inferred from patterns in the sequences [4]. The result of this pre-trained protein model and a fingerprinted compound are then passed into various machine learning models to predict CPI or binding affinity for each compound-protein pair.

This project aims to explore and compare these two state-of-the-art techniques (BACPI and a particular language model named ESM-2). We hope you enjoy it.

Please note that our final presentation, codes, paper, and poster can be found here.

Proteins

Proteins have diverse biological functions ranging from DNA replication, acting as enzymes and hormones, providing nutrient transport, making antibodies, enabling wound healing and tissue regeneration, transporting oxygen around the bodies of multicellular organisms, converting one molecule into another and many others. In fewer words, they underpin every aspect of biological activity. This is particularly important in areas where protein structure and function have an impact on human endeavour such as medicine [6].

Amino acids are the building blocks of proteins; there are twenty different amino acids, all of which are required to make the many different proteins found in the human body − a single protein is composed of a unique sequence of covalently linked amino acids.

A protein's primary structure is the linear order of AA residues. The conformation of the amino acids forms the protein's secondary structure: α-helices and β-sheets. The α-helix is formed when the primary chain coils to form a spiral structure, which is stabilized by hydrogen bonds and the β-helix is formed when the primary chain "zig-zags" to form a "pleated" sheet where adjacent strands are held together by hydrogen bonds. A protein's tertiary structure is formed when α-helices or β-sheets fold to form a compact globular molecule held together by intramolecular bonds. The quaternary structure forms when two or more polypeptide chains, each with its own tertiary structure, combine to form a functional protein [6].

A binding site is a small pocket in a protein's *tertiary structure* that allows for binding to an incoming molecule also known as a ligand (this project refers to a ligand as a compound). A single protein may have more than one binding site. In the context of drugs, the amount of drug that binds determines its effectiveness. A bound drug is kept in the bloodstream while an unbound drug is metabolized or excreted [5]. These binding sites are critical because the

binding of a molecule to a protein often triggers a change in conformation in the protein and results in altered cellular function [6]. Therefore, a protein's three-dimensional structure can be a key to understanding its function [5]. We aim to represent each compound-protein pair's three-dimensional structure and their potential binding sites, predict if there is an interaction and the strength of the interaction between the compound-protein pair.

Approach

3.1 Datasets

In binary classification-based CPI prediction studies, compound–protein pairs are labeled as positive or negative samples. Often, experimentally validated CPIs are treated as positive samples and the unconfirmed CPIs are treated as negative samples. However, these random negative samples may include unknown positive samples and *that* would greatly influence the accuracy and credibility of the model. Therefore, screening true negative samples is necessary to construct credible CPI datasets. To evaluate the classification performance of our model in CPI prediction task, we use human and C.*elegans* CPI datasets created by Liu et al. (2015). The positive samples of the datasets were collected from DrugBank 4.1 (Wishart et al., 2008), Matador (Gunther et al., 2008) and STITCH 4.0 (Kuhn et al., 2014). It contains 3369 positive interactions for human datasets and 4000 positive interactions for C.*elegans* datasets. The negative samples were retrieved by a screening framework. There are 384,916 negative samples for the human datasets and 88,261 negative samples for the C.*elegans* datasets [3].

In regression-based compound–protein binding affinity prediction studies, compound–protein pairs are labeled as binding affinity values. There are four measures of binding affinity: IC_{50}, K_i, K_d and EC_{50}. In this project, we use four datasets created by Karimi et al. (2019) to evaluate our models for this prediction task. The IC_{50} dataset contain 489,280 samples, the K_i dataset contains 144,525 samples, the K_d dataset contains 12,589 samples, and the EC_{50} dataset contains 37,896 samples. Note that we used binding affinity in its logarithmic form (i.e. $-log_{10}IC_{50}$) as the target label. Each dataset is also split into a training and test set [2, 3].

All datasets contain three columns of information in the following order: the SMILES representation of a compound, the AA sequence of the target protein, and either the target interaction value (0 for a negative interaction or 1 for a positive interaction) or target binding affinity value for that particular compound-protein pair.

3.2 BACPI Model

The BACPI portion of our project uses features obtained by a GAT and fingerprinting representation of compounds and a CNN for representations of proteins. For the compounds, we use RDKit to convert each SMILES format into a graph representation and employ the GAT to extract information (such as atom types, aromaticity and chemical bond types) from the graph. For the proteins, our CNN takes each protein's AA sequence and learns the feature representation of the protein. Finally, an attention-based bi-directional neural network is used to integrate the representations of compounds and proteins and predict the interaction and binding affinity of the input compound–protein pair [3]. Please see figure 6.1 in Appendix A.

3.2.1 Compound Representation

Graph Attention Layer (GAT)

A GAT allows for assigning different weights to different nodes within a neighborhood, instead of the mean over all neighbors' representation vectors. It can also learn and update representation vectors for every atom of compound by iteratively gathering information from the neighbors of each atom, so that each single atom extracts the local feature of the compound substructures [3].

First, we used RDKit to convert the SMILES format of a compound to its graph representation, where a graph was represented as $G = \{V, E\}$, where V is the set of vertices and each vertex $v_i \in V$ represents the ith atom, and E is the set of edges and $e_{ij} \in E$ is the

chemical bond between the ith atom and jth atom. Then, G and the randomly initialized embeddings of vertices are fed into the GAT. Note that we represented the embeddings of vertices as $v = \{v_1, v_2, ..., v_{N_v}\}$ where $v_i \in \mathbb{R}^{\mathbb{H}}$, where N_v is the number of vertices or atoms of vertices (i.e. atoms) and H_c is the dimension of the vertex embedding. A linear transformation was applied to increase the expressive power and transform the embeddings into higher-level features. The attention coefficient a_{ij} indicated the importance of the jth vertex features to the ith vertex, and it was computed using:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\overrightarrow{a}^T[\overrightarrow{v_i}'||\overrightarrow{v_j}']))}{\sum_{k\in N_i}\exp(\text{LeakyReLU}(\overrightarrow{a}^T[\overrightarrow{v_i}'||\overrightarrow{v_k}']))} \tag{3.1}$$

where N_i are the neighbors of the ith vertex, α is the shared attention mechanism parameterized by a weight vector $\overrightarrow{a} \in \mathbb{R}^{2H_c'}$ and $||$ is the concatenation operation. Then, we used the normalized coefficients to update the vertex hidden vector which is the final output features for each vertex:

$$\overrightarrow{v_i}' = \sigma\left(\sum_{j\in N_i}\alpha_{ij}\overrightarrow{v_j}'\right) \tag{3.2}$$

We also extended the mechanism to a multi-headed attention by combining the output features of K independent graph attention layer to stabilize the learning process. So:

$$\overrightarrow{v_i}^{cat} = \Big\Vert_{k=1}^{K}\sigma\left(\sum_{j\in N_i}\alpha_{ij}^k\overrightarrow{v_j}'^k\right) \tag{3.3}$$

Here, $||$ is the concatenation operation, $\overrightarrow{v_i}^{cat} \in \mathbb{R}^{KH_c}$ is the concatenation of K layer, and α^k are the normalized attention coefficients of the kth layer. Lastly, a single-layer neural network was used to transform $\{\overrightarrow{v_i}^{cat}\}_{i=1}^{N_v}$ into a compound space:

$$\overrightarrow{v_i} = \text{LeakyReLU}(W_{\text{out}}\overrightarrow{v_i}^{cat}) \tag{3.4}$$

where $W_{\text{out}} \in \mathbb{R}^{H_c x K H_c'}$ is a transition parameter and $\overrightarrow{v_i} \in \mathbb{R}^{H_c}$ is one of the atom's features

[3].

Fingerprinting

We also used extended connectivity fingerprints (ECFPs) to describe the molecular characteristics of each atom in each compound. We used RDKit to calculate the fingerprint of compounds and obtain a feature vector $\overrightarrow{f}$ of length 1024, which was transformed into the compound space by a multi-layer neural network [3]. Our final features for each compound included the atom, the adjacency measure between each atom, and a radius analysis.

3.2.2 Protein Representation

The input for each protein is its AA sequence and a CNN is used to extract its features and supply their vector representations. First, to capture all potential subsequences, we used a context window w to split the protein sequences into overlapping subsequences of amino acids. Since there are 20 types of amino acids, the total possible subsequences for each protein is 20^w. In this project, we set the context window $w = 3$. For example, given an input of a protein sequence MRPSG...FIGA, this is split into the subsequences 'MRP', 'RPS', 'PSG',..., 'FIG', 'IGA', each of which are regarded as AA residues [3].

Now, Let $S = \{s_1, s_2, ..., s_{L_p}\}$ be a protein's AA sequence, where s_i is the ith amino acid and L_p is the length of the protein sequence. First, the protein sequence S was split into a residue sequence $R = \{r_1, r_2, ..., r_N\}$ where r_i is the ith residue and N_r is the number of residues and equals $L_p - w + 1$. Then, we translated all residues to randomly initialized embeddings $R = \{\overrightarrow{r_1}, \overrightarrow{r_2}, ..., \overrightarrow{r_N}\}$ and $\overrightarrow{r_i} \in \mathbb{R}^{H_p}$. Here, H_p is the dimension of residue embeddings. These randomly initialized embeddings are updated through several convolutional layers with a non-linear activation function (ReLU). The hyperparameters of our CNN include the number of convolutional layers, the size and the number of filters. Lastly, we obtained the output $\{\overrightarrow{r_i} \in \mathbb{R}^{H_p}\}_{i=1}^{N_r}$ for all residues in each protein's sequence [3].

3.2.3 CPI & Binding Affinity Prediction

Once we obtained the feature representations of the compounds and proteins, we used a bi-directional attention neural network that produces attentions in two directions: atom-to-residue and residue-to-atom, for each compound and protein pair to enable each compound–protein pair to be aware of each other [3].

First, the compound's atom features $\{\overrightarrow{v_i}\}_{i=1}^{N_v}$ and fingerprint features $\overrightarrow{f}$ and the protein's residue features $\{\overrightarrow{r_i}\}_{i=1}^{N_r}$ were each transformed into vectors with same dimension d by single-layer neural network:

$$\overrightarrow{c_i} = \text{LeakyReLU}(\mathbf{W}_v \overrightarrow{v_i}) \tag{3.5}$$

$$\overrightarrow{h_f} = \text{LeakyReLU}(\mathbf{W}_f \overrightarrow{f}) \tag{3.6}$$

$$\overrightarrow{p_i} = \text{LeakyReLU}(\mathbf{W}_r \overrightarrow{r_i}) \tag{3.7}$$

where $W_v \in \mathbb{R}^{d \times H_c}$, $W_f \in \mathbb{R}^{d \times H_c}$, and $W_r \in \mathbb{R}^{d \times H_c}$, d is the dimension in this bi-directional attention neural network module, and $\{\overrightarrow{c_i}\}_{i=1}^{N_v}$, $\overrightarrow{h_f}$, and $\{\overrightarrow{p_i}\}_{i=1}^{N_v}$ are the respective vectors of the transformed atom, fingerprint, and residue features.

If $C = [\overrightarrow{c_1}^T, \overrightarrow{c_2}^T, ..., \overrightarrow{c_{N_v}}^T] \in \mathbb{R}^{N_v \times d}$ is the context matrix of a given compound and $P = [\overrightarrow{p_1}^T, \overrightarrow{p_2}^T, ..., \overrightarrow{p_{N_r}}^T] \in \mathbb{R}^{N_r \times d}$ is the context matrix of a given protein, their alignment matrix can be calculated using

$$A \in \mathbb{R}^{N_v \times N_r} = A\text{tanh}(CUP^T) \tag{3.8}$$

where $U \in \mathbb{R}^{d \times d}$ is a trainable parameter matrix and A is a pairwise interaction matrix and A_{ij} is the interaction strength between the ith atom and jth residue. We extracted the information weighted by A for each atom of the compound for all residues of the protein and

each residue of the protein by

$$\mathbf{I}_c = \mathbf{A}\tanh(\mathbf{P}\mathbf{W}_{r2a}) \tag{3.9}$$

$$\mathbf{I}_p = \mathbf{A}^T\tanh(\mathbf{C}\mathbf{W}_{a2r}) \tag{3.10}$$

where $\mathbf{W}_{r2a}$, $\mathbf{W}_{a2r} \in \mathbb{R}^{d\times d}$, $\mathbf{I}_c \in \mathbb{R}^{N_r\times d}$ is the information delivered from atoms to residues. Then, the attentions in both directions, atom-to-residue α_{a2r} and residue-to-atom α_{r2a} are calculated using:

$$\alpha_{a2r} = \text{softmax}([\mathbf{C}\mathbf{W}_c||\mathbf{I}_c]\alpha_{a2r}) \tag{3.11}$$

$$\alpha_{r2a} = \text{softmax}([\mathbf{P}\mathbf{W}_p||\mathbf{I}_p]\alpha_{r2a}) \tag{3.12}$$

where W_c, $W_p \in \mathbb{R}^{d\times d}$, $||$ is the concatenation operation, $\alpha_{a2r} \in \mathbb{R}^{2d}$ and $\alpha_{r2a} \in \mathbb{R}^{2d}$ are the attention mechanisms of atom-to-residue and residue-to-atom, and $\alpha_{a2r} \in \mathbb{R}^{N_v}$ and $\alpha_{r2a} \in \mathbb{R}^{N_r}$ are the compound and protein attentions normalized by the softmax function [3].

Finally, the compound features $\overrightarrow{h}_c$ and the protein features $\overrightarrow{h}_p$ are the weighted sums of atom features and residue features based on the respective normalized attentions α_{a2r} and α_{r2a}:

$$\overrightarrow{h}_c = \alpha_{a2r}\cdot\mathbf{C} \tag{3.13}$$

$$\overrightarrow{h}_p = \alpha_{r2a}\cdot\mathbf{P} \tag{3.14}$$

where $\overrightarrow{h}_c$, $\overrightarrow{h}_p \in \mathbb{R}^d$ [3].

We stabilized the learning process of the bi-directional attention by extending the attention mechanism to multi-head attention. Specifically, L independent bi-directional attention mechanisms execute equations 8 to 14 to obtain different compound and protein features, which are then each concatenated [3]. Fully expanded, the concatenated compound features $\overrightarrow{h_c}^{cat}$ and protein features $\overrightarrow{h_p}^{cat}$ were expressed as:

$$\overrightarrow{h_c}^{cat} = \Big\Vert_{l=1}^{L} \text{softmax}([\mathbf{C}\mathbf{W}_c^l || \mathbf{A}^l \tanh(\mathbf{P}\mathbf{W}_{r2a}^l)]\alpha_{a2r}^l) \cdot \mathbf{C} \tag{3.15}$$

$$\overrightarrow{h_p}^{cat} = \Big\Vert_{l=1}^{L} \text{softmax}([\mathbf{P}\mathbf{W}_p^l || \mathbf{A}^l \tanh(\mathbf{C}\mathbf{W}_{a2r}^l)]\alpha_{r2a}^l) \cdot \mathbf{P} \tag{3.16}$$

$$\mathbf{A}^l = \tanh(CU^l P^T) \tag{3.17}$$

where $\overrightarrow{h_c}^{cat}$, $\overrightarrow{h_p}^{cat} \in \mathbb{R}^{Ld}$. These concatenated features are then transformed into dimension d by a single-layer neural network resulting in the final representation of the compound's and protein's features:

$$\overrightarrow{h_c}^{final} = \mathbf{W}_{fc}\overrightarrow{h_c}^{cat} \tag{3.18}$$

$$\overrightarrow{h_p}^{final} = \mathbf{W}_{fp}\overrightarrow{h_p}^{cat} \tag{3.19}$$

where $\mathbf{W}_{fc}$, $\mathbf{W}_{fp} \in \mathbb{R}^{d \times Ld}$ and $\overrightarrow{h_c}^{final}$, $\overrightarrow{h_p}^{final} \in \mathbb{R}^d$. At long last, the CPI or binding affinity is predicted using:

$$\gamma = \mathbf{W}_\gamma \text{LeakyReLU}(\text{flatten}([\overrightarrow{h_c}^{final} || \overrightarrow{h_f}] \otimes \overrightarrow{h_p}^{final})) \tag{3.20}$$

where the concatenation of the compound feature $\overrightarrow{h_c}^{final}$ and fingerprint $\overrightarrow{h_f}$ is the representation of the overall compound feature, $\otimes$ is the outer product, and the outer product of the

compound and protein representation is flattened into a column vector of length $2d^2$ [3].

3.3 Evolutionary Scale Model (ESM)

The ESM is a language model. In artificial intelligence, text based language models that fill in missing words or predict the next word in a sentence or phrase have been shown to develop "thinking" that appears to be connected to the underlying meaning of the text. However, the machine is merely regurgitating patterns that it developed as a result of computations, data, and increasing parameters and training examples [4]. This can be applied to proteins because words bear relations and "interact with their neighbors" in the same way amino acids depend on their sequential surroundings. Moreover, the detrimental effect of adding/changing a letter/word/phrase can alter a sentence's meaning is equivalent to changing an amino acid in a sequence [1]. For these reasons, we chose ESMs to represent the proteins in our datasets as part of a second approach to predict CPIs and binding affinity.

For clarity, the proteins are pre-trained with an ESM-2 model and the compounds are fingerprinted using RDKit. Then, these results are passed into various supervised and unsupervised machine learning models to predict the CPI or binding affinity for a compound-protein pair.

3.3.1 Compound Representation

Once again, we used fingerprinting to describe the molecular characteristics of each atom in each compound. We used RDKit to calculate the fingerprint of compounds and obtain a feature vector $\overrightarrow{f}$. Our final features for each compound included the atom, the adjacency measure between each atom, and a radius analysis.

3.3.2 Protein Representation

First, an ESM-2 language model was created for each protein in a compound-protein pair in our datasets. Those ESM-2 language models were trained − at scales of 8 million to 15 billion parameters − to predict the identity of randomly selected amino acids in a protein sequence by observing their context in the rest of the sequence causing the model to learn the dependencies between the amino acids [4]. In plainer words, the input is the original amino acid sequence. The model randomly removes amino acids and this new input is fed into a BERT-style encoder only transformer and via self-supervised learning, the model learned how to predict the missing amino acids thereby gaining insight into the structure of the protein. The output is the three-dimensional coordinates and confidences of the binding sites of the protein.

3.3.3 CPI & Binding Affinity Prediction

Once we obtained the feature representations of each compound-protein pair, we applied various supervised and unsupervised models to those results to predict either the CPI interaction or binding affinity.

3.4 Models & Evaluation

We used the following models for our project:

- BACPI (bi-directional attention neural network)
- ESM (Evolutionary Scale Modeling − specifically, the ESM-2 model)
    - Supervised Models
        - Classification
            - Linear Regression

            - Elastic Net Regression
            - Random Forest
            - Extra Trees
            - SVM
            - Gradient Boosted
            - GaussianNB
            - Multinomial NB
            - Logistic Regression
            - Perceptron
            - Multi-layer Perceptron
            - KNN
        * Regression
            - Ridge Linear Regression
            - Lasso Linear Regression
            - SVM (Gaussian RBF)
            - Random Forest
            - Multi-layer Perceptron
    - Unsupervised Models
        * Affinity Propogation
        * KMeans

We also implemented Stochastic Gradient Descent to find the hyperparameters needed to optimize our models.

We evaluated our models using the Root Mean Squared Error (RMSE).

Results & Discussion

The Appendix contains three tables that show the current results for our project. Before we begin to dissect those tables, please note that we were unable to gather results for the CPI task for the BACPI model and therefore cannot compare the BACPI and ESM models for the CPI task.

For the CPI prediction − the prediction of whether or not there is an interaction between a compound-protein pair − of the ESM models, the Extra Trees model performed the best for the C. *elegans* and human datasets with an RMSE of 0.189474 and 0.228321. The worst performing model for the C. *elegans* dataset is Linear Regression with an RMSE in the trillions (1778409083) and for the human dataset is KMeans with an RMSE of 0.834972822.

So, why may the Extra Trees model be the best model (so far, until we have BACPI results)? Well, the Extra Trees algorithm works by creating a large number of unpruned decision trees from the dataset. In the case of classification, predictions are made using majority voting from the decision trees. Random Forest also gave good RMSE results − it came in second best for both the C. *elegans* and human datasets. Unlike the Random Forest algorithm that develops each decision tree from a bootstrap sample of the training dataset, the Extra Trees algorithm fits each decision tree on the whole training dataset.

Linear Regression was by far the worst for the C. *elegans* dataset but not the human dataset, which is curious. Both datasets have the same type of data (i.e. SMILES data for the compound, amino acid sequences for the protein, and the target value). We know that linear regression works well when the following conditions are met: a linear relationship between the independent and dependent variables, independent residuals, homoscedasticity, and residuals have a normal distribution. Perhaps, one of these conditions is not being met for the C. *elegans* dataset. At this time, we suspect that there does not exist a linear

relationship between the compound-protein pair.

For the binding affinity prediction − the prediction of the binding strength between a compound-protein pair − BACPI performed the best across all of the datasets for which we have data. The only other model that came close is the ESM Random Forest model. Studies have shown that decision trees like the Random Forest model works well for tabular and structured data (vision and audition) with *small sample sizes* while neural networks perform better on structured data with *larger sample sizes*.

The predicted values for the interaction and binding affinity were both continuous values and not probabilistic, so the RMSE was the best choice to analyze the results. In the future, if our models can be modified for the results to be discrete (either 0 or 1) for the classification task, we would then be able to output a confusion matrix, which would give us an exact number of false positives and false negatives *for a particular compound-protein pair*. Those numbers are important because the last thing we want is to claim that there is or is not an interaction between a compound-protein pair, when the opposite is true. It would be interesting to see which model outputs the lowest false positives and/or false negatives − it may not be the extra trees model that currently has the lowest RMSE. Another point to note is if the same compound-protein pairs are giving false negatives and false positives for each model. Perhaps, there is not enough structural training data for that pair.

We previously stated that we implemented SGD to find the optimal hyperparameters for our model. Those hyperparameters are {'alpha': 0.1, 'eta0': 0.0001, 'l1_ratio': 0.2, 'learning_rate': 'invscaling', 'max_iter': 100, 'penalty': 'l1'}. However, the RMSE for this model is also absurdly high (486037551.49), so we opted not to implement any of these hyperparameters. We also removed the adjaceny feature for the compounds to lesson the run-time of our codes and we found that it did not make much difference with regards to the RMSE.

Lastly, we were also under time constraints and were unable to deep-dive into the issues mentioned above nor modify our original codes.

Conclusion

The identification of CPIs is an essential step in drug discovery. Because the experimental determination of CPIs is both expensive and time-consuming, the computational model offers a promising and efficient alternative. We explored two state-of-the-art techniques: a bi-directional attention neural network named BACPI and a language model named ESM-2 for predicting these interactions and the binding affinity between a compound-protein pair. Although our experiment is incomplete, so far we found that the ESM-2 Extra Trees model was the best for predicting an interaction and the BACPI model was best for predicting binding affinity.

Lessons Learned re: Machine Learning

Machine learning can take something as simple as filtering ham and spam emails and create a solution that can be automated efficiently and at scale. However, a machine learning model cannot just work on anything that is thrown at it without some kind of external guidance. This project made us realize how thorough and expansive this technology is, but a great deal of research still needs to be done on our part.

Challenges & Future Work

We attempted to code and thoroughly understand a very large project in a small amount of time. Apart from the time constraints, scope of the work, and sloppy planning, it never occurred to us that it would take multiple days (even weeks) to run a single code. As a result, we have plenty of unfinished work. To properly compare the BACPI and various ESM models, we would like to run the BACPI codes on the human dataset, execute the various supervised and unsupervised ESM models on the datasets with the adjacency feature included for for

the compounds, and finally run the ESM binding affinity codes on the EC_{50} dataset. We would also like to take a deeper look into why hyperparameter tuning did not work for our models and checking the correctness of our work with additional evaluation parameters (i.e. r^2, confusion matrix, AUC, ROC, etc.).

Additional future work also includes testing various other ESM models from Meta, testing ESM with a neural network for classification and binding affinity instead of supervised and unsupervised models, and pre-training our proteins with AlphaFold instead of ESM.

Acknowledgements

The authors would like to say thank you to Hunter College for admitting us into their graduate program and Professor Anita Raja for her guidance and never-ending support. We are very grateful.

Bibliography

[1] Ferruz, Noelia & Höcker, Birte. (2022). Towards Controllable Protein design with Conditional Transformers.

[2] Karimi, M., Wu, D., Wang, Z., & Shen, Y. (2019). *DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks*. Bioinformatics (Oxford, England), 35(18), 3329–3338. https://doi.org/10.1093/bioinformatics/btz111.

[3] Min Li, Zhangli Lu, Yifan Wu, and YaoHang Li. *BACPI: a bi-directional attention neural network for compound–protein interaction and binding affinity prediction*. Bioinformatics, 38(7), March 2022, pp. 1995–2002, https://doi.org/10.1093/bioinformatics/btac035.

[4] Lin, Zeming, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, et al. "Evolutionary-Scale Prediction of Atomic Level Protein Structure with a Language Model." bioRxiv, 2022. https://doi.org/10.1101/2022.07.20.500902.

[5] Toutain, P. L.; Bousquet-Melou, A. (2002-12-14). *Free Drug Fraction vs. Free Drug Concentration: A Matter of Frequent Confusion*. Journal of Veterinary Pharmacology and Therapeutics. Wiley inc. 25 (6): 460–463. https://doi.org/10.1046/j.1365-2885.2002.00442.x.

[6] Whitford, David. 2013. Proteins: Structure and Function. J. Wiley & Sons.

Appendix A

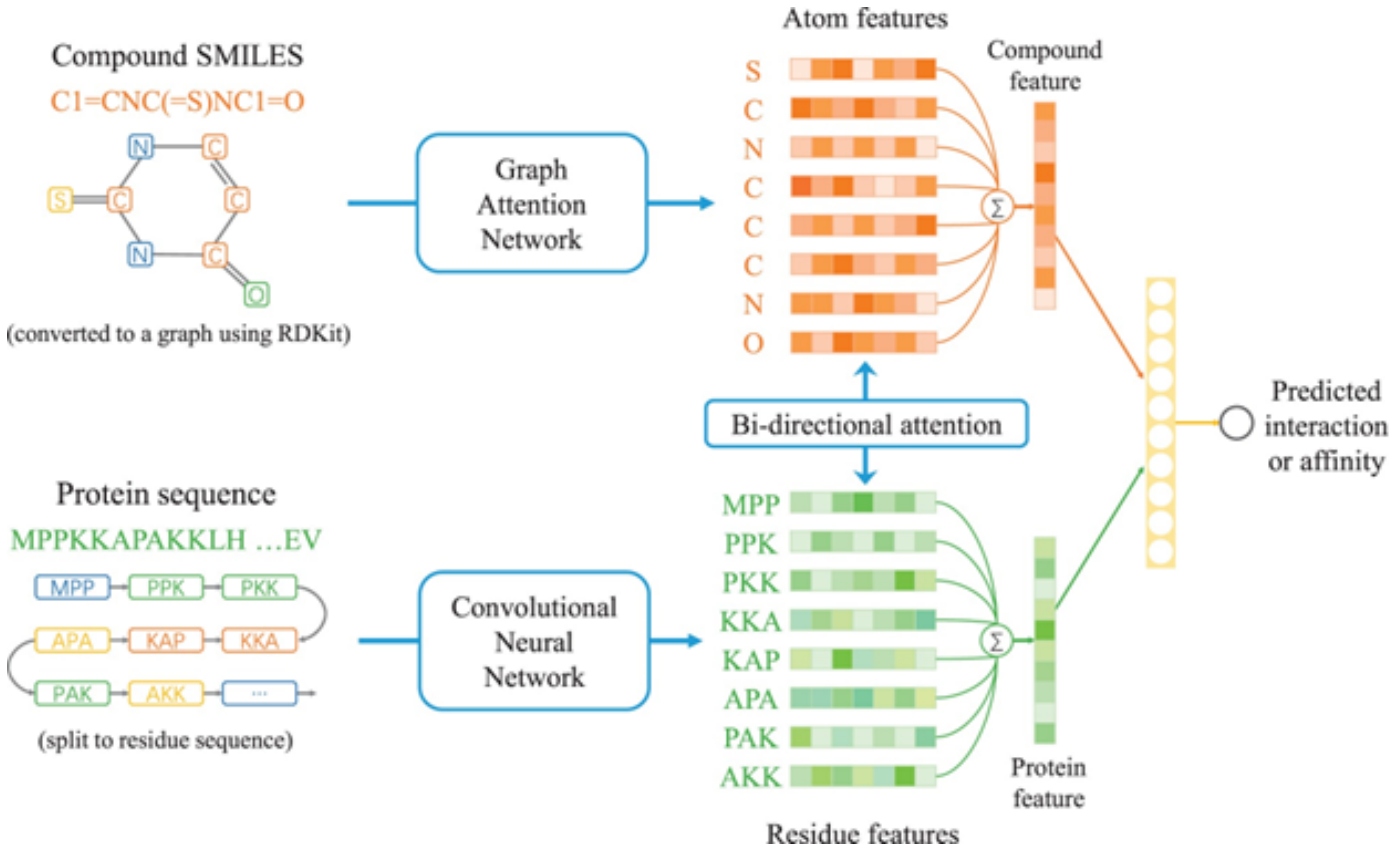

Figure 6.1: An image of the BACPI process [3].

Appendix B

C. elegans	
Model Name	**RMSE**
Extra Trees	**0.189474**
Random Forest	0.211261
Gradient Boosted	0.273941
SVM	0.297722
Logistic Regression	0.322886
KNN	0.360259953
ElasticNet Regression	0.372044
Multi-Layer Perceptron	0.407379
GaussianNB	0.409982
Local Outlier Factor	0.50318407
MultinomialNB	0.518796
Perceptron	0.705601
Kmeans	0.777395731
Linear Regression	1778409083
Spectral Clustering	-
Affinity Propagation	-

Table 7.1: CPI (classification) results for the C. *elegans* dataset.

Human	
Model Name	**RMSE**
Extra Trees	**0.228321**
Random Forest	0.242715
SVM	0.291737
Gradient Boosted	0.307586
Logistic Regression	0.339053
Linear Regression	0.363745
ElasticNet Regression	0.372946
KNN	0.385137992
Perceptron	0.408501
Multi-Layer Perceptron	0.408501
MultinomialNB	0.455701
Local Outlier Factor	0.496150446
GaussianNB	0.777464
Kmeans	0.834972822
Spectral Clustering	-
Affinity Propagation	-

Table 7.2: CPI (classification) results for the human dataset.

Model Name	IC50	EC50	Ki	Kd
BACPI	**0.74**	**0.78**	**0.8**	**1.08**
Random Forest	0.792437	-	0.989753	1.184066
MLP Regressor	0.917291	-	1.083637	1.259279
Ridge Linear Regression	0.949984	-	3.034943	2.285426
Lasso Linear Regression	1.013284	-	1.422459	1.367706
Support Vector Machine (Gaussian RBF)	1.186531	-	1.198874	1.421241

Table 7.3: Binding affinity results for the BACPI and ESM models.