Protein-Ligand Interaction Classification Using Machine Learning Models

Course: CB&B 750 (SP24): Core Topics in Biomedical Informatics

Professor Samah Fodeh-Jarad

Group 12: Jiacheng Zu, Zhongzheng Mao, Wei Feng Siew, Xinyi Guo, Shuangrui Chen

Abstract

The classification of protein-ligand interactions is crucial for advancing our understanding of biological processes and enhancing drug discovery efforts. Traditional methods, which rely heavily on manual feature extraction and expert analysis, face challenges in terms of efficiency and scalability. This study addresses these limitations by building different machine learning models to automate the classification process. Utilizing advanced representation learning techniques, specifically ProtVec and SMILES2Vec, complex protein sequences and chemical structures were transformed into informative, low-dimensional vectors. Advanced algorithms—support vector machines, logistic regression, and random forests—were employed and evaluated on a dataset containing 36,000 protein-ligand pairs, ensuring a balanced representation of positive and negative samples. The performance of these models was analyzed, with the support vector machine model demonstrating the highest levels of precision and recall. The findings underscore the potential of machine learning to significantly improve the accuracy of protein-ligand interaction predictions and point towards the integration of more complex molecular features to further enhance classification performance.

1. Introduction

Protein-ligand interactions are fundamental to numerous biological processes including signal transduction, cellular regulation, and immune responses [1-3]. These interactions, driven by the physicochemical properties of both proteins and ligands, are crucial for the modulation and effective functioning of biological pathways. However, the prediction of these interactions remains a formidable challenge due to the variability in protein structures and the complex nature of their binding mechanisms [4,5].

Traditionally, experimental methods such as X-ray crystallography and NMR spectroscopy have been employed to study protein-ligand interactions. While these approaches are effective, they are also time-consuming and expensive [6]. Consequently, there has been a significant shift towards computational methods, which promise faster and cost-effective predictions. Among these, machine learning (ML) and deep learning techniques have emerged as powerful tools capable of deciphering complex biological data [7].

Despite considerable advancements in computational approaches, accurately predicting whether a specific protein and ligand pair will interact and form a stable complex remains a critical issue. Many existing models struggle with robustness and accuracy, particularly when faced with the diverse and dynamic nature of molecular interactions [8].

This research focuses on applying different advanced machine learning techniques to classify protein-ligand interactions, leveraging a rich dataset of protein-ligand complexes. By employing state-of-the-art representation learning methods like ProtVec and SMILES2Vec, and integrating them into powerful classifiers such as support vector machines, logistic regression, and random forests, this study aims to surpass traditional feature extraction methods in both efficiency and predictive accuracy.

The following sections will detail the methodologies used in developing these models, present the comparative analysis of their performance, and discuss the implications of our findings for the field of bioinformatics. This introduction sets the stage for understanding how machine learning can revolutionize the classification of protein-ligand interactions, shifting the paradigm from labor-intensive manual methods to automated, data-driven approaches.predicting protein-ligand bindings, followed by a discussion of the implications of these findings and potential areas for future research.

2. Literature review

Various machine learning methods have been developed in the last decades. Due to the complexity of protein structures, the performance of machine learning methods relies heavily on data representation (or features). Therefore, the design of data preprocessing and data transformation is of great concern to ensure that the data representation can support efficient machine learning algorithms. However, these feature extraction methods require tremendous manpower and expert insights. Representation Learning aims to automatically learn the representations (or features) from raw data that can be effectively utilized by downstream machine learning models to improve the performance of the model. For example, Word2vec is one of the most popular RL methods. Word2vec has been adapted to protein sequences (ProtVec) for classification of protein families and prediction of disordered Proteins. The SMILES2Vec is a model that introduces a direct conversion of chemical structures from SMILES (Simplified Molecular-Input Line-Entry System) strings into vectors. These works show that RL technologies represented by Word2vec can automatically learn low-dimensional features from compound and protein feature space and achieve excellent performances. SPVec vectors were constructed via the combination of SMILES2Vec and ProtVec to represent specific interactions.

3. Methods

3.1 Dataset

In this study, the dataset, Pdb_protein_ligand_complexes is employed, which is a public, web-accessible dataset from huggingface [9]. The whole dataset claimed to contain about 36,000 unique pairs of protein sequences and ligand SMILES, and the coordinates of their complexes from the PDB. SMILES are assumed to be tokenized by the regex from P. Schwaller.

3.2 Machine Learning Models

Zhongzheng Mao's work primarily utilizes Python to construct machine learning models for the study, with an emphasis on leveraging libraries such as Pandas, NumPy, and Scikit-learn. The initial analysis of the dataset indicated that all samples were proteins known to bind specific ligands, inherently labeled as positive. To address the lack of negative samples, a method was devised to generate them by randomly selecting pairs of protein-ligand complexes (distinguished by their pdb_ids) and exchanging their ligand-related data, including coordinates (ligand_xyz, ligand_xyz_2d), bonding information (ligand_bonds), and SMILES notation. Initially, a dictionary was constructed containing protein-ligand pairs known to interact (positive samples). Subsequently, negative samples were created by randomly selecting a protein-ligand pair not present in the dictionary and appending their attributes to a new row in the dataset. This method ensured the generation of an equivalent number of negative samples, thereby balancing the dataset. Owing to the extensive computational demands of processing the entire dataset, a subset was randomly selected for model training to enhance efficiency.

For feature selection, Mao chose the protein sequences (Seq) and the SMILES strings of the ligands. These features were treated analogously to text in natural language processing, where each amino acid or chemical bond is regarded as a distinct "word". The use of TF-IDF (Term Frequency-Inverse Document Frequency) with settings of analyzer='char' and ngram_range=(1, 3) allowed for the extraction of single characters, bigrams, and trigrams as features, capturing local structural patterns. This conversion of sequences and SMILES strings into numerical vectors via TfidfVectorizer facilitated their use in supervised learning models. Given the binary classification nature of the task, suitable algorithms such as support vector machines, logistic regression, and random forests were employed to construct the predictive models. These methods were selected for their robustness and suitability in handling binary data outcomes.

Similar to Mao, Wei Feng Siew's work utilizes Python to construct machine learning models for the study by leveraging libraries such as Pandas, NumPy, and Scikit-learn. The only difference between Mao and Siew's work is the method used to generate vector representations of protein-ligand pairs. First, the same method mentioned by Mao was used to include negative samples in the dataset (i.e. protein-ligand pairs not known to interact), creating a class-balanced dataset with 8000 positive and 8000 negative samples. Next, a modified version of the SPVec method was followed to create vector representations of proteins and ligands. Specifically, protein Seq strings and ligand SMILES strings in the dataset were split into tokens, treating every 3 consecutive characters as a token. Two separate Word2Vec models were then trained, one using the tokenized Seq strings and the other using tokenized SMILES strings as input. The following hyperparameters were used to train the models: skip-gram approach (sg=1), negative sampling (negative = 5) and window_size =5. From each trained Word2Vec model, vector representations of tokens were extracted. To construct the vector representation of a single protein or ligand, the vector representations of constituent tokens were first summed up, and then divided by the number of constituent tokens, yielding the mean vector representation of constituent tokens. Next, for each protein-ligand pair, Siew concatenated the vector representation of the protein with the vector representation of the ligand, yielding the vector representation of a protein-ligand pair. Finally, vector representations of protein-ligand pairs and their binding ability (0 for non-binding and 1 for binding) were used to train predictive models based on various algorithms such as support vector machine, logistic regression, and random forest.

4. Results

To predict protein-ligand binding, we proposed three models, implemented by Xinyi Guo & Shuangrui Chen, Zhongzheng Mao, and Wei Feng Siew respectively. In Xinyi Guo & Shuangrui Chen's implementation, many problems were encountered, including path issues, memory problems, etc. Specifically, some errors occurred due to "Module Not Found" or "Incorrect Working Directory", leading to a failure in running the code. Moreover, it appears that vector extraction requires a large amount of memory, such that the computers (MacBook) were not capable of processing the data and they crashed. Another interesting problem was shown as "SystemExit: 2" with an error message "ipykernel_launcher.py: error: unrecognized arguments". However, the indicated argument is generally used by Jupyter internally to specify a connection file which should not be provided manually, but should rather be handled automatically by the Jupyter infrastructure. Due to the issues mentioned above, Xinyi Guo & Shuangrui Chen were not able to successfully implement their prediction model.

In Zhongzheng Mao's contribution to the study, which utilized TfidfVectorizer vector representations, the effectiveness of the machine learning models was primarily evaluated using four performance metrics: accuracy, precision, recall and F1 score. The performance metrics obtained from 5-fold cross validation indicate that none of the three machine learning models examined are optimal, although the support vector machine demonstrates marginally superior performance. In observing the performance metrics of the random forest model—namely precision, recall, and F1 score, all of which are zero—it can be inferred, based on the mathematical formulations of these indices, that the model classified almost every test data instance with a negative label. This suggests that the model failed to learn the appropriate patterns or associations between proteins and ligands, indicating a significant shortfall in its predictive capability.

	Average accuracy	Average precision	Average recall	Average F1 score
SVM	0.667	0.600	0.002	0.004
Logistic Regression	0.659	0.075	0.002	0.004
Random Forest	0.666	0.000	0.000	0.000

Next, in Wei Feng Siew's contribution to the study, which utilized SPVec vector representations, the effectiveness of machine learning models was evaluated using performance metrics obtained from 5-fold cross validation. All performance metrics indicate that the support vector machine model yields the best performance, followed by random forest, and then logistic regression. Moreover, in terms of average precision, average recall and average F1 score, all models trained on SPVec vector representations displayed overwhelmingly superior performance relative to those trained on TfidfVectorizer vector representations.

	Average accuracy	Average precision	Average recall	Average F1 score
SVM	0.570	0.562	0.637	0.596
Logistic Regression	0.479	0.480	0.490	0.484
Random Forest	0.556	0.553	0.597	0.573

5. Discussion

In Zhongzheng Mao's segment of the research, the approach initially involved using only two features to construct the models, but there was an attempt to incorporate four additional attributes: receptor_features, ligand_xyz, ligand_xyz_2d, and ligand_bonds. The strategy was to transform these attributes into feature vectors using methods such as flattening and scaling, and then vertically stacking these vectors using the np.vstack function to create a 2D array. However, this process was fraught with challenges including the occurrence of NaN values, inconsistent lengths of feature vectors among samples, and irregular shapes in the feature matrix.

Furthermore, issues such as negative indices arose, possibly due to variations in protein sequences and differences in the chemical structures of ligands, which result in differing dimensions and types of feature vectors, complicating the processing. These difficulties highlight the complex nature of the data and suggest areas for further investigation in future studies.

Another significant challenge faced was the computational demand; processing the full dataset proved to be excessively time-consuming, potentially taking several days. Consequently, a subset of the data was randomly sampled for model training, which, while necessary to manage time constraints, could affect model performance. Moreover, it was observed that the models occasionally predicted extreme outcomes, either suggesting that all proteins could bind to ligands or none could, indicating a failure to adequately learn the underlying patterns between proteins and ligands. Although the models' average accuracy were over 60%, the other three indices, average precision, average recall and average F1 score were quite low. This is likely due to the inadequacy of TfidfVectorizer vector representations for the purpose of training machine learning models. By training models on SPVec vector representations, Wei Feng Siew's segment of the research yielded models with overwhelmingly superior performance relative to those trained on TfidfVectorizer vector representations. This underscores the importance of using suitable vector representations so as to yield machine learning models with superior predictive capabilities.

6. Conclusion

ML methods rely heavily on data representation (or features). However, traditional feature extraction methods require tremendous manpower and expert insights, and the effectiveness of these features also requires tremendous computations to be proved. [10] Representation learning (RL) is a way to introduce artificial intelligence (AI) and prior knowledge to automatically learn continuous, information-rich and lower dimensional vectors from raw data that can be easily and directly used in ML models. Our work has shown that by integrating SPVec methods in vector representations, we have successfully improved the accuracy of our machine learning model in predicting protein ligand bindings.

Our limitation worth mentioning is a lack of negative samples for training machine learning models. Additionally, the attempt to include additionally attributes including receptor_features, ligand_xyz, ligand_xyz_2d, and ligand_bonds will require further study to solve problems caused by the occurrence of NaN values, inconsistent lengths of feature vectors among samples, and irregular shapes in the feature matrix. As future work, we plan to include the physicochemical properties of proteins and ligands to improve the feature extraction. With better vector representations, our ML-based prediction models will be able to reach a higher level of accuracy.

Bibliography

[1] Du, X., Li, Y., Xia, Y. L., Ai, S. M., Liang, J., Sang, P., ... & Liu, S. Q. (2016). Insights into protein–ligand interactions: mechanisms, models, and methods. International journal of molecular sciences, 17(2), 144.

[2] Pujadas, G., Vaque, M., Ardevol, A., Blade, C., Salvado, M. J., Blay, M., ... & Arola, L. (2008). Protein-ligand docking: A review of recent advances and future perspectives. Current Pharmaceutical Analysis, 4(1), 1-19.

[3] Zhao, L., Zhu, Y., Wang, J., Wen, N., Wang, C., & Cheng, L. (2022). A brief review of protein–ligand interaction prediction. Computational and Structural Biotechnology Journal, 20, 2831-2838.

[4] Dhakal, A., McKay, C., Tanner, J. J., & Cheng, J. (2022). Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. Briefings in Bioinformatics, 23(1), bbab476.

[5] Colwell, L. J. (2018). Statistical and machine learning approaches to predicting protein–ligand interactions. Current opinion in structural biology, 49, 123-128.

[6] Vajda, S., & Guarnieri, F. (2006). Characterization of protein-ligand interaction sites using experimental and computational methods. Current Opinion in Drug Discovery and Development, 9(3), 354.

[7] Colwell, L. J. (2018). Statistical and machine learning approaches to predicting protein–ligand interactions. Current opinion in structural biology, 49, 123-128.

[8] Zhao, L., Zhu, Y., Wang, J., Wen, N., Wang, C., & Cheng, L. (2022). A brief review of protein–ligand interaction prediction. Computational and Structural Biotechnology Journal, 20, 2831-2838.

[9] Glaser, J. (2024). PDB protein ligand complexes. Hugging Face.

https://huggingface.co/datasets/jglaser/pdb_protein_ligand_complexes

[10] Zhang, Y.-F., Wang, X., Kaushik, A. C., Chu, Y., Shan, X., Zhao, M.-Z., Xu, Q., & Wei, D.-Q. (2019, December 12). SPVec: A word2vec-inspired feature representation method for drug-target interaction prediction. Frontiers.

https://www.frontiersin.org/articles/10.3389/fchem.2019.00895/full#B1

Project Division

Name	Presentation Contribution	Report Contribution
Jiacheng Zu	Introduction, Data, Workflow	Abstract, Introduction, Format Editor
Zhongzheng Mao	Methods, Results	Methods, Results, Discussion
Wei Feng Siew	Methods, Results	Methods, Results
Xinyi Guo	Literature Review, Problems Met, Discussion	Introduction, Results, Conclusion
Shuangrui Chen	Literature Review, Problems Met, Discussion	Introduction, Results, Conclusion

All of us contributed equally to the project in various roles.

Sign: Jiacheng Zu, Zhongzheng Mao, Wei Feng Siew, Xinyi Guo, Shuangrui Chen