# Classifying Protein-Metabolite interactions within the Micro-Biome using Machine Learning

Jack Gregorski

Princeton University

April 28, 2025

**Abstract**

*Understanding interactions between microbial metabolites and human immune proteins is important for understanding the role of the micro-biome in health and disease. In this work, I develop a machine learning framework that combines protein language model* (PLM) *embeddings with molecular fingerprints to predict protein–small molecule interactions. The dataset was constructed from the* STITCH *and* STRING *databases, and protein clustering and synthetic negative generation were used for balanced training. A threshold-based evaluation strategy was introduced to assess model robustness across varying confidence levels. The findings demonstrate the promise of these models for identifying new interactions between the micro-biome and the immune system. Limitations regarding experimental validation and potential errors are discussed.* **Code available at:** `https://github.com/JackGregorski/IW.git`

# 1 Introduction

At a young age, the human immune system begins to interact with the bacteria that live within us called the micro-biome. At the cellular level, these interactions are governed through the transmission of small chemical molecules that are produced by the bacteria, and then bind to receptor proteins on the immune cells and trigger certain reactions. Understanding the interactions between these microbial small molecules, called metabolites, and human immune system proteins is very important for advancing research areas like therapeutics, micro-biome research, and immunology. Small molecules that are produced by bacteria can affect inflammation, autoimmune issues, and disease progression. However, our ability to predict which of these compounds will interact with specific proteins is still limited. For example, recent studies suggest that microbial metabolites play roles in diseases like Crohn's disease and rheumatoid arthritis. The underlying molecular interactions are often unknown [1] [2]. Despite the growing interest in micro-biome-host interactions, systematic prediction of protein-chemical interactions in this context remains an open challenge.

The problem that I am to solve or at least contribute to here is to predict, given a bacterial small molecule and a human immune protein, whether the two will interact. Large databases like STITCH and STRING have cataloged some known interactions, they are biased toward well-studied organisms and chemicals, and experimental discovery is time-consuming and costly [3] [4]. There are existing machine learning methods that either focus on human-human protein interactions, or they predict chemical binding broadly without accounting for the unique properties of micro-biome-derived molecules or immune proteins. Because of

this, I think there is a lot of research and improvement in tools that are used for this area.

For this project, I used a machine learning-based approach that takes advantage of pre-trained protein language models and molecular fingerprints to build a classifier specifically designed for predicting micro-biome-small molecule and immune protein interactions. My approach is new in three key ways: (1) it focuses on bacterial metabolites which is a less researched area; (2) it utilizes embeddings from state-of-the-art protein language models rather than traditional feature engineering; Finally, (3) it systematically benchmarks performance across different data thresholds and compares results against logistic regression and dummy classifiers. This adds important context by providing a scalable, efficient tool for hypothesis generation in micro-biome-immunology research. This may be able to offer new insights where there is little experimental data.

# 2 Background and Related Work

A central challenge in drug discovery is the efficient and accurate prediction of protein-small molecule interactions. Traditional experimental approaches, such as surface plasmon resonance (SPR), isothermal titration calorimetry (ITC), and X-ray crystallography, provide high-precision validation of binding but are costly and time-consuming [5]. This inspired the development of computational methods to predict protein-chemical interaction. These methods are broadly categorized into structure-based and sequence-based approaches.

With structure-based methods like ScanNet, P2Rank, and DeepSite the three-dimensional structural information of proteins is processed in order to predict ligand-binding pockets.

This often requires extremely time-intensive structure preparation or inference using advanced tools like AlphaFold [5]. These methods are highly accurate, but their approaches are often not scalable to large protein screenings.

In contrast, sequence-based approaches offer a more accessible pipeline because they require only the amino acid sequence of a protein, rather than its full 3D structure. Amino acids are the chemical "building blocks" that link together in chains to create proteins. Each protein can be represented as a linear sequence of amino acids. This sequence can encode important information about the protein's properties and structure. Recently, pretrained protein language models (PLMs) have become more widely utilized and are powerful tools to represent proteins in a biologically informative way. Singh et al. introduced ConPLex, a deep learning architecture that combines PLM-based encodings with contrastive learning to coembed proteins and small molecules in a shared space for interaction prediction [6]. This allows ConPLex to generalize across low- and zero-shot drug-target interactions (DTI) scenarios while maintaining high specificity, especially when distinguishing between true binders and decoy molecules. Unlike many sequence-based models that rely solely on labeled DTI pairs, ConPLex leverages PLM representations pretrained on millions of protein sequences. This enhances the model's generalizability and allows it to overcome training data scarcity.

A complementary line of work focuses on binding site prediction instead of direct interaction classification. Wang et al. proposed CLAPE-SMB, a model that integrates a pretrained PLM with contrastive learning to predict small molecule binding sites on proteins, even in intrinsically disordered regions [5]. This method demonstrates that contrastive learning is not only valuable in interaction prediction but also in identifying functionally important

regions in proteins.

Beccaria et al. took a structure-based approach that combines invariant molecular representations and graph neural networks (GNNs) to predict binding based on protein pocket geometry [7]. Their work emphasizes physical realism and interpretable representations for proteins. Their model achieves competitive results through non-deep learning classifiers that generalize well despite being trained on empirically limited data.

In summary, recent advances use deep-level representations of proteins, specifically pre-trained language models and GNNs, which allow them to overcome the limitations in traditional binding prediction methods. My work expands upon this by extending predictive modeling to the domain of micro-biome-immune system interactions, a space that has mostly been ignored by previous DTI methods. I also combine protein language model embeddings with molecular fingerprints to represent proteins and small molecules respectively. This enables generalization in both the protein and chemical fields. Additionally, we develop a novel threshold-based evaluation framework that systematically tests the trained models across varying confidence levels of known interactions. This is used to highlight model robustness in sparse and noisy biological settings, filling a void that has not been investigated by the existing models.

# 3 Approach and Implementation

## 3.1 Data Collection and Preprocessing

The central dataset used for this study was the STITCH dataset, which catalogs protein-chemical interactions [3]. To narrow the biological scope and reduce the volume of interactions, only human proteins were retained, focusing the study on interactions within the human micro-biome. This filtering yielded a total of 15,473,940 interacting pairs.

Each interaction pair in the dataset consists of a protein and a chemical and is accompanied by a confidence score. This score is calculated using a naive Bayesian combination of multiple evidence sources. These sources include experimental data, database annotations (from KEGG and DrugBank), prior predictions, and literature text mining. The combination formula is [3]:

$$\text{Score}_{\text{total}} = 1 - \prod_i (1 - p_i)$$

This naive approach, however, tends to overestimate the interaction probability when many low-confidence sources are included. To mitigate this, a correction based on the prior probability of observing interactions is applied by STITCH [3].

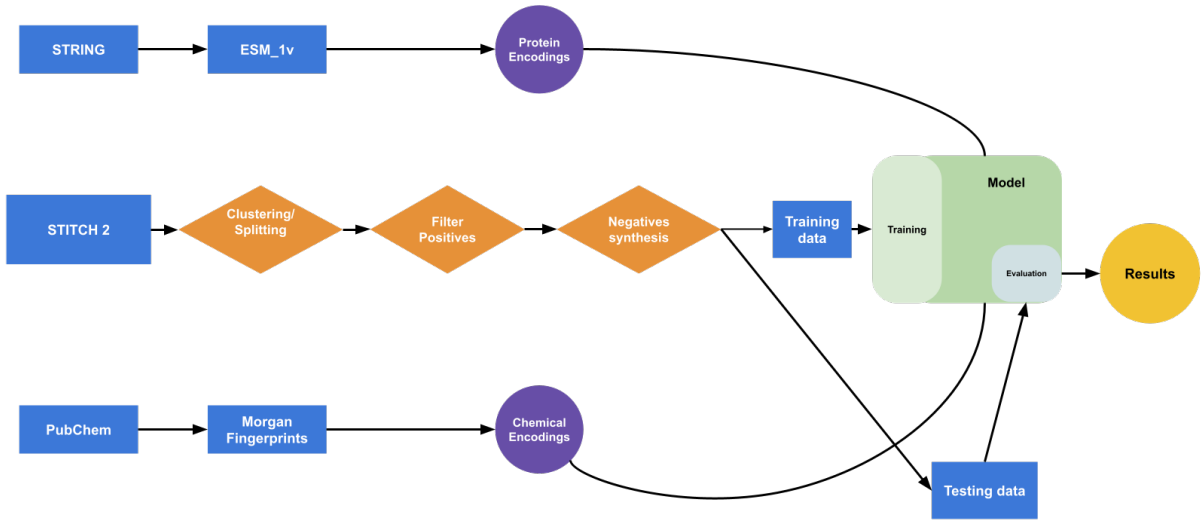


Figure 1: Overview of Data Processing and Model Training Pipeline. Protein sequences from STRING are embedded using ESM-1v, while chemical structures from PubChem are processed into Morgan fingerprints. Interaction pairs from STITCH are filtered by confidence threshold, clustered to prevent data leakage, and supplemented with synthesized negatives. The resulting training and testing datasets are used to train and evaluate the predictive model, with results analyzed across multiple evaluation metrics.

## 3.2 Molecular Fingerprint Generation

A key step in my process was generating molecular fingerprints to represent the chemicals in a format that the model could understand. Fingerprints are fixed-length binary vectors that encode the presence or absence of specific chemical substructures. They are widely used in computational chemistry for tasks such as similarity search, clustering, and predictive modeling.

The STITCH database provides a supplemental dataset containing all chemical names and SMILES strings. SMILES (Simplified Molecular Input Line Entry System) are simple, human-readable strings that encode a molecule's structure using ASCII characters. For example, "CCO" represents ethanol [3].

The full STITCH chemical dataset had over 116 million entries, my first step was to filter these down to only the chemicals relevant to this experiment. I extracted a list of the 786,495 unique chemicals in the interaction-pairs dataset and then processed the chemical database to filter to just these.

Using the SMILES strings, I generated molecular fingerprints via the RDKit library. Specifically, I used the Morgan fingerprinting method. This encodes circular substructures around each atom within a defined radius. For each chemical, a 2048-bit binary fingerprint vector was generated, where each bit represents the presence or absence of a specific substructure [8].

These fingerprint vectors were stored in a lookup table indexed by chemical ID, allowing for fast access during model training and inference. By putting the chemical structure information into a numerical format, I then allowed my model to learn complex patterns of interaction between chemicals and proteins.

## 3.3 Protein Embedding Generation

I downloaded a set of *Homo sapiens* proteins and their corresponding amino acid sequences from the STRING database [4]. Then, to reduce the size of this dataset and to make sure it aligned with our interaction dataset, I filtered this set to only include proteins that appeared in our STITCH 2 dataset. This results in a final set of 19,196 unique proteins.

To convert protein sequences into vector representations usable by our model, we used a pretrained Protein Language Model(PLM) known as ESM-1v, created by Meta [9]. PLMs

operate in a similar way to natural language models. They treat amino acid sequences as sentences and individual amino acids as tokens, which allows them to learn very detailed representations through unsupervised training on massive protein databases. These embeddings capture structural and functional properties of proteins—including shape, chemical properties, and evolutionary information [9].

Each protein sequence was passed through the ESM-1v model to obtain a fixed-length embedding vector. This results in a vector with 1280 components that captures all the structural and chemical properties of the protein [9].

We stored these vectors in a lookup table indexed by unique protein IDs. During training and testing, the model retrieves these embeddings as needed to pair with chemical fingerprints. By using a pretrained PLM, we avoided the need to train protein representations from scratch and leveraged generalized knowledge learned from millions of protein sequences which improves generalization across different proteins.

## 3.4 Protein Clustering and Data Splitting

To prevent data leakage and to get an accurate measure of model performance, proteins were clustered by sequence similarity prior to splitting the dataset. Data leakage in this setting can occur when closely related proteins, those with highly similar sequences or structural features, appear in both the training and test sets. Protein embeddings capture biologically meaningful patterns, such as evolutionary or functional similarity. This means that a model trained on one protein might indirectly recognize a related protein during testing

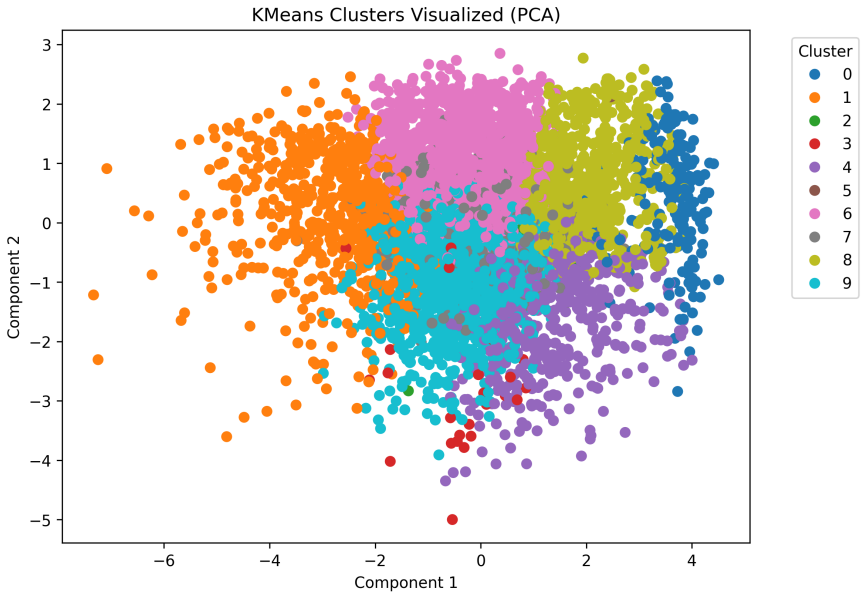


Figure 2: Protein clusters formed using k-means (k=10) after reducing protein embeddings to two dimensions via PCA for visualization.

even without having seen it directly. This leads to overfitting, where the model performs well not because it has generalized, but because it has memorized features specific to certain protein families. Such leakage can result in inflated performance metrics that do not reflect the model's actual ability to predict interactions for new and unseen proteins. By clustering proteins and assigning entire clusters exclusively to either the training or testing set, this risk is substantially reduced. This allows for a more reliable assessment of model generalization.

To address this, we applied k-means clustering to the protein embeddings with $k = 1000$. Among the 238,607 interaction pairs, there were 19,000 unique proteins. This resulted in clusters of approximately 19 proteins on average.

Clusters were then used to stratify the dataset into training and testing sets, ensuring that no cluster appeared in both splits.

Table 1: Number of Protein-Chemical Pairs in Training and Testing Sets Across Thresholds

| | **Threshold** | | | | | |
|---|---|---|---|---|---|---|
| **Set** | **400** | **500** | **600** | **700** | **800** | **900** |
| Train | 1,712,323 | 996,797 | 780,115 | 462,557 | 266,265 | 82,769 |
| Test | 402,615 | 330,065 | 152,035 | 139,285 | 62,497 | 29,991 |

## 3.5 Threshold-Based Data Splitting

After clustering proteins and dividing the dataset into separate training and testing groups based on cluster assignment, the data was further organized by interaction confidence thresholds. Specifically, I assigned a series of confidence thresholds for each pair (e.g., 400, 500, 600, 700, 800, 900). The training and testing data was then filtered into new datasets based on these thresholds, so for each threshold, there was a corresponding training and testing set containing only interacting pairs with a confidence interval greater than the corresponding threshold value.

This two-step filtering process, first filtering by protein cluster and then by threshold—was important for maintaining consistency across model comparisons. Splitting data based on clusters before applying the threshold filter ensured that no closely related proteins appeared in both training and testing sets across any threshold. This eliminated potential sources of data leakage while allowing each thresholded model to be trained and tested on interaction sets specific to different levels of confidence.

Doing this also allowed models trained on different training thresholds to be directly compared to each other via using the same training set. This ensures that differences in the observed performance are related to the specific model's accuracy and confidence and not

due to inconsistencies in data splitting.

## 3.6 Generating Negative Samples

Since the STITCH dataset lacks labeled negative interactions, we synthetically generated negatives by randomly pairing chemicals and proteins that do not appear together anywhere in the dataset. These random pairs were assumed to be non-interacting and were labeled accordingly. Due to significant variability in the frequency of protein occurrences across the dataset, I implemented a dynamic negative sampling strategy in which the number of negative examples per protein was matched to its corresponding count of positive interactions. This ensured class balance on a per-protein basis, helping to reduce potential bias during model training.

## 3.7 Model Architecture and Training

After preparing the embeddings and filtering invalid entries, we created training examples by concatenating chemical fingerprints and protein embeddings as input vectors. Each pair was labeled as either 1 (positive) or 0 (negative).

We used a Feedforward Neural Network (FNN) architecture composed of multiple fully connected layers with dynamically decided activation functions and dropout layers in between to prevent overfitting.

To optimize the model, we used Optuna for hyperparameter tuning. The search space included:

- Number of hidden layers: 1–3
- Hidden layer sizes: 64–512 (step size 64)
- Dropout: 0.2–0.5
- activation function: ReLU, Leaky ReLU(0.1), GELU
- l2 regularization: $10^{-6}$ to $10^{-3}$
- Learning rate: $10^{-5}$ to $10^{-2}$
- Batch size: 32 or 64
- Epochs: 5–15

Each Optuna trial used a 3-fold group cross-validation split via GroupKFold, where protein clusters defined the groups. This made sure that no protein cluster was shared between training and validation folds.

The final model was trained on the full training set using the best hyperparameters and evaluated with early stopping based on validation loss.

## 3.8 Evaluation

To assess model performance, we used a diverse set of evaluation metrics that capture different aspects of binary classification quality:

- **Accuracy** measures the overall proportion of correct predictions but can be misleading in the presence of class imbalance.

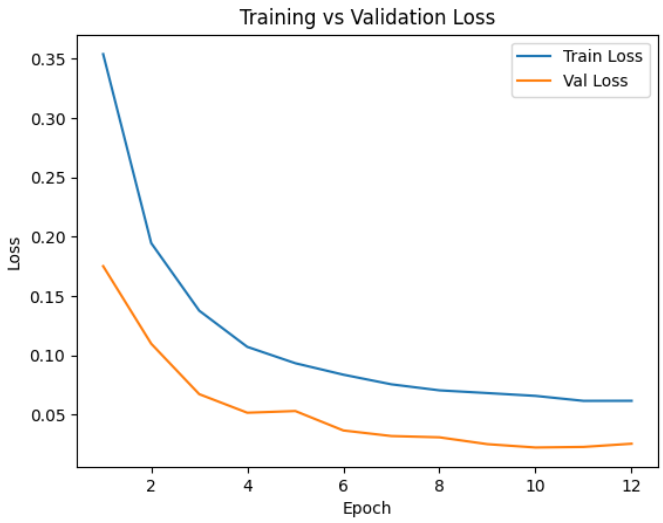


Figure 3: Training and validation loss curves for threshold 900. Validation loss is lower because dropout is only being applied during training.

- **Precision** evaluates how many of the predicted positive interactions were actually correct, while **Recall** measures how many of the actual positives were successfully identified.

- **F1-score** provides a harmonic mean of precision and recall, and is especially useful when the classes are imbalanced.

- **AUROC (Area Under the Receiver Operating Characteristic Curve)** captures the model's ability to rank positive examples higher than negatives, regardless of threshold.

- **Brier Score** quantifies the accuracy of probabilistic predictions, penalizing both overconfident incorrect predictions and underconfident correct ones.

- **Log Loss** evaluates the log-likelihood of the predicted probabilities, emphasizing confident mistakes more heavily than Brier Score.

We interpret these metrics jointly to develop a well-rounded understanding of the models' behaviors. For example, high AUROC paired with low precision might suggest that the model ranks examples well, but its classification threshold needs adjustment.

In addition to evaluating our primary neural network model, we also compare its performance to two baseline models:

1. A **Logistic Regression** classifier trained on the same concatenated input features, serving as a linear benchmark.

2. A **Dummy Classifier** that always predicts the majority class, providing a naive baseline for performance under extreme class imbalance.

These baselines help contextualize the learned model's results: strong performance relative to both confirms the added value of the learned representations and nonlinear decision boundaries. Visual comparisons between these models (e.g., via ROC) are included in the Results section.

# 4 Results

## 4.1 Criteria for Success

There were two primary success criteria for this work. First, aim to develop models that significantly outperform baselines like logistic regression by distinguishing unique relationships in interacting protein-chemical pairs as measured by AUROC. The second success criterion

was to develop models that can perform robust generalization across datasets with different interaction strength thresholds and show signs of minimal overfitting to a specific training dataset or threshold. Successful generalization is defined here as the model's ability to be trained on one threshold and still be able to achieve high AUROC and low Brier scores (the primary metrics used in my evaluation) on the unseen test sets at different confidence thresholds, specifically those representing stronger interaction confidence, i.e higher thresholds.

## 4.2 Model Performance Overview

The main evaluation metric was AUROC. Across all models and testing thresholds, the highest recorded AUROC was achieved by the model trained at threshold 600, evaluated on the threshold 900 testing set, with an AUROC of **0.989**. This result indicates extremely strong discernibility in identifying true interactions at the highest test set threshold.

Logistic regression, used as a baseline model, achieved its best performance on the threshold 900 dataset with an AUROC of **0.79**. Every trained neural network model outperformed this logistic baseline across all thresholds. This demonstrates the effectiveness of our learned feature representations. A notable result is that models consistently performed worst when evaluated on test sets that matched their own training threshold. I hypothesize that although proteins were clustered to minimize data leakage, models may have overfit to threshold-specific characteristics. Minor data leakage, such as imperfect separation of homologous proteins across splits, could also contribute to this result.

Models tended to perform better when tested on thresholds higher than their training

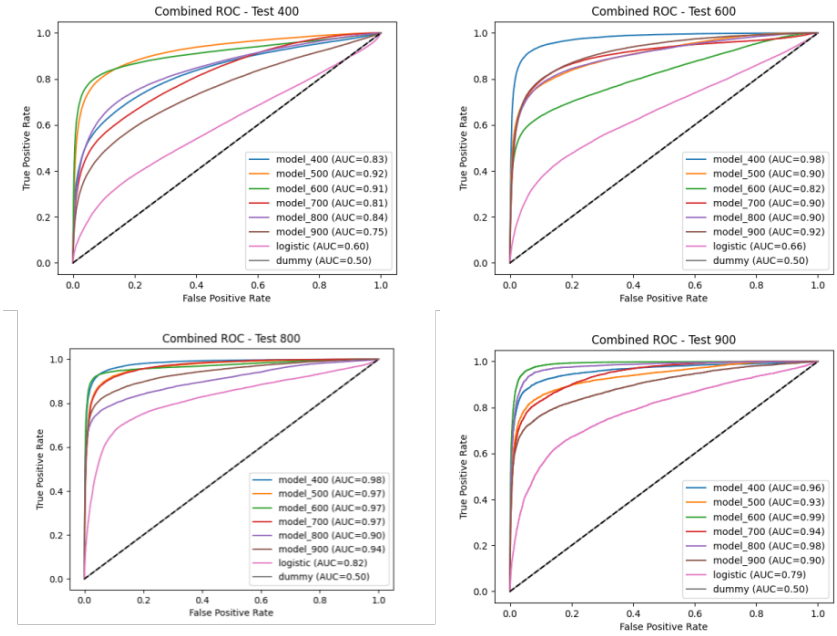


Figure 4: **Combined ROC Curves for Model Performance Across Different Test Set Thresholds.** ROC curves are shown for models trained at thresholds 400, 500, 600, 700, 800, and 900 when evaluated on four different test sets (thresholds 400, 600, 800, and 900). Each model's area under the curve (AUC) is reported in the legend. Logistic regression and dummy classifiers are included as baselines. Models trained at lower thresholds generally demonstrate strong cross-threshold generalization, with particularly high AUCs observed on the higher-threshold test sets

threshold and worse on thresholds lower than their training threshold. This behavior is what I expected, as higher thresholds represent "easier" examples of strong/likely interactions, while lower thresholds include weaker, noisier interactions not represented during training. Furthermore, positives at higher thresholds are subsets of those at lower thresholds, meaning models trained at higher thresholds lacked exposure to some positives that appeared only in lower-threshold datasets. These trends can be clearly seen in the AUROC heatmap (see Figure 5). Models exhibit the lowest AUROC when evaluated on their own threshold, slightly better AUROC on lower thresholds, and significantly higher AUROC on thresholds above their own training threshold.


Figure 5: Heatmap showing AUROC scores across training and testing thresholds. Rows represent models trained at a given threshold; columns represent evaluation thresholds. Models generally perform better on higher-threshold test sets than on their own or lower-threshold test sets.

## 4.3 Secondary Metrics

Brier scores further corroborated the AUROC findings. The lowest Brier score, indicating the most accurate probability calibration, was **0.037**. This result was achieved by the model trained at threshold 600 on the threshold 900 test set. A histogram of prediction probabilities (see Figure 7) illustrates that these predictions were tightly concentrated near 0 and 1, which reflects high model confidence.

Precision and recall metrics followed similar patterns to AUROC. Heatmaps (Figures 8) demonstrate that precision and recall were lowest when evaluated on the model's own training threshold, improved on lower thresholds, and were highest when evaluated on thresholds

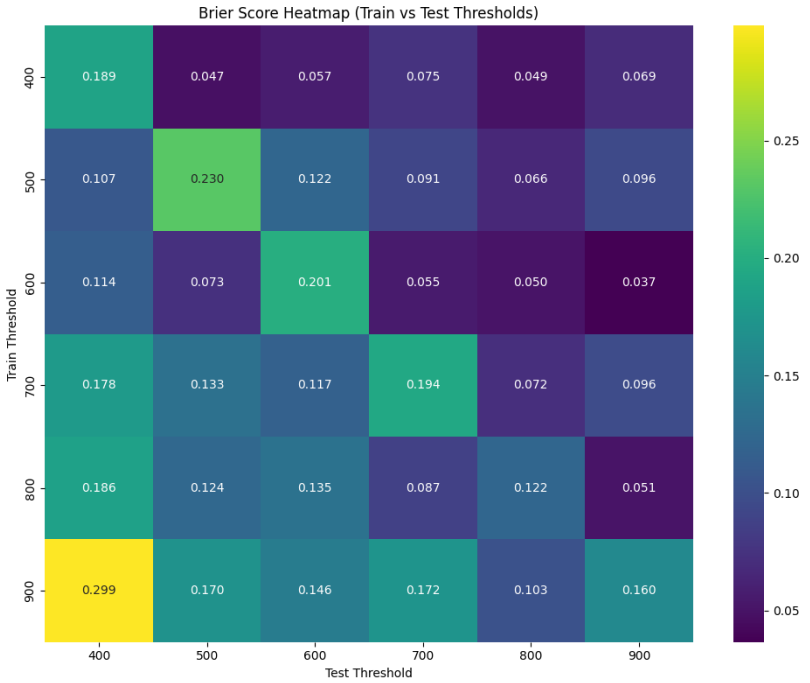


Figure 6: **Brier Score Heatmap Across Training and Testing Thresholds.** Each cell shows the Brier score achieved when a model trained at a given threshold (y-axis) is evaluated on a test set with a specific threshold (x-axis). Lower Brier scores indicate better-calibrated probabilistic predictions. Models trained at lower thresholds (400–600) generally achieve stronger calibration across a range of test thresholds, particularly at higher thresholds.

higher than the training threshold.

# 5 Conclusion and Future Work

## 5.1 Conclusion

The findings I arrived at the end of this work demonstrate significant promise in the application of machine learning to identify interactions between proteins and small molecules in the micro-biome. In particular, the use of combining protein language model encoding with molecular fingerprints to vectorize the data provided an extremely effective form of feature

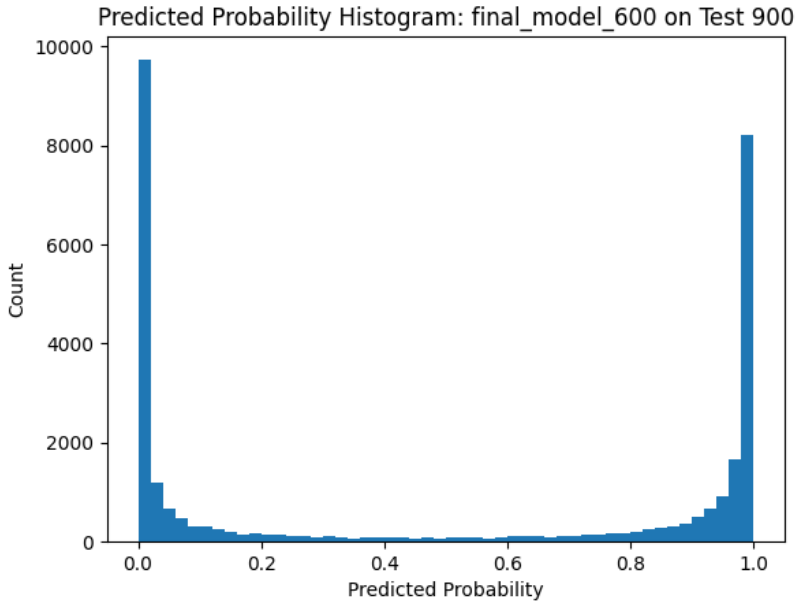


Figure 7: **Predicted Probability Histogram for Model Trained at Threshold 600 Evaluated on Test Set 900.** The histogram shows the distribution of predicted probabilities output by the model trained at threshold 600 when evaluated on the threshold 900 test set. The strong peaks on the far range indicate that most predictions clustered near 0 or 1 which indicates high model confidence in classifying interactions as either positive or negative.

representation for classification tasks.

The best-performing model was trained at threshold 600 and evaluated on higher threshold test sets. This model achieved an accuracy of **0.954**, an AUROC of **0.989**, a Brier score of **0.037**, an F1 score of **0.954**, a log loss of **0.134**, a precision of **0.952**, and a recall of **0.955**. These high performance metrics suggest that machine learning approaches, when properly structured, offer strong feasibility for predicting protein-small molecule interactions, particularly within complex biological systems such as the human micro-biome.

I had several important findings as a result of these experiments. First, training models on lower thresholds appeared to enhance their ability to generalize across unseen datasets.

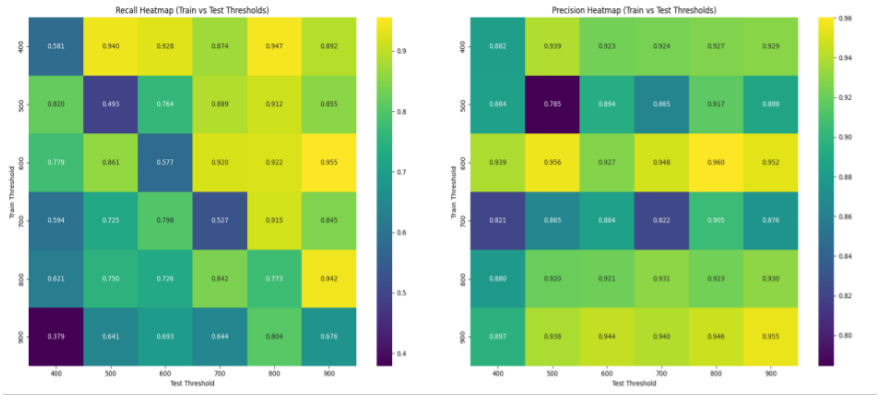


Figure 8: **Recall and Precision Heatmaps Across Training and Testing Thresholds.** The left heatmap shows recall values, and the right heatmap shows precision values for models trained at thresholds 400–900 and evaluated on test sets with corresponding thresholds. Higher values indicate better performance. Models trained at lower thresholds (400–600) generally achieve strong recall and precision across a range of test thresholds, suggesting good generalization and balanced prediction behavior.

This may be due to the fact that lower thresholds provide greater access to a wider variety of interaction examples, including weaker or noisier interactions. Exposure to these "harder" cases may have encouraged the model to learn more fundamental features associated with binding. However, models trained on too low a threshold, such as 400, underperformed relative to the model trained at threshold 600. This suggests that extremely low thresholds may introduce confusing or mislabeled examples representing very weak or rare interactions, which ultimately worsen model performance.

In comparing these results to previous experiments done in this area, the very high AUROC and accuracy achieved by the best model significantly exceed reported baselines. This offers more evidence that the machine learning techniques used here can be a viable and powerful tool for identifying novel protein–small molecule interactions in the micro-biome. Singh et al. [6] introduced ConPLex, a contrastive learning model built on protein language models, achieving AUROC scores around 0.87–0.90 in low-coverage settings. Beccaria et al. [7] used

an invariant molecular fingerprinting strategy paired with random forest classifiers, obtaining AUROCs of approximately 0.81 on challenging datasets such as DUD-E. In contrast, the best model in this study achieved an AUROC of **0.989** on thresholded test sets, showing a much stronger performance. However, it is important to note that these prior works addressed very different tasks, used different datasets, and addressed different problems, and therefore direct performance comparisons cannot be fruitfully made. Furthermore, as my analysis has not been validated, the results I report here could contain errors or overestimate the real performance. Despite this, the findings here support the proposal that combining molecular fingerprints and protein embeddings, even with simple feedforward architectures and threshold tuning strategies, can achieve competitive or cutting-edge predictive power for identifying biologically significant protein-small molecule interactions.

Overall, our results show that combining carefully selected training thresholds with well-represented features allows for high predictive performance in micro-biome protein-small molecule interaction prediction. This study highlights the importance of threshold selection and data quality, and supports the growing role of computational methods in advancing biological discovery. It also provides a starting point for future research building on machine learning-guided chemical screening.

## 5.2 Future Work

There are several avenues through which this work can be expanded and improved.

First, although clustering was used to minimize data leakage between proteins in the

training and test sets, no analogous measures were applied to the chemical space. Future work could focus on clustering or partitioning chemical fingerprints to be more careful and potentially prevent chemical-based data leakage, which may have contributed to models' overfitting their training threshold.

Second, the feature representations themselves could be improved. Incorporating additional protein encoding models, or using more chemically-informative representations for small molecules such as chemical embeddings generated from graph neural networks could enhance model performance.

Further analysis could also be conducted to study cluster-specific performance. For example, investigating which protein clusters exhibited the lowest classification accuracy may uncover interesting patterns that could provide insight into how the model classifies interactions. Similarly, examining whether particular families of proteins or chemicals are consistently misclassified could provide insight into the limitations of the current representations or model structure.

Aside from model refinement, more biological validation is important, especially testing the model on different datasets. Future studies could apply the trained model to the MiMeDB dataset, which catalogs microbial metabolites [10]. By first extracting potential interacting pairs between MiMeDB molecules and known targets from the STITCH dataset, and then evaluating model performance on this expanded micro-biome-specific dataset, we could assess the model's utility in a real-world application setting. A similar evaluation could be performed using the STRING database [4] filtered to human immune system proteins, providing an opportunity to test predictive accuracy against a high-priority set of targets.

If model performance on these new datasets remains high, it would open the door to using machine learning classifiers for large-scale chemical screenings, identifying novel protein-metabolite interactions relevant to human health and disease.

# 6 Acknowledgments

I would like to acknowledge Professor Mona Singh as well as Sam Neff for their invaluable guidance and assistance in completing this project.

# 7 Honor Code

This paper represents my own work in accordance with Princeton University regulations.

- Jack Gregorski

# References

[1] Yu-Ling Chang, Maura Rossetti, Hera Vlamakis, David Casero, Gemalene Sunga, Nicholas Harre, Shelley Miller, Romney Humphries, Thaddeus Stappenbeck, Kenneth W. Simpson, R. Balfour Sartor, Gary Wu, James Lewis, Frederic Bushman, Dermot P.B. McGovern, Nita Salzman, James Borneman, Ramnik Xavier, Curtis Huttenhower, and Jonathan Braun. A screen of crohn's disease-associated microbial metabo-

lites identifies ascorbate as a novel metabolic inhibitor of activated human t cells. *Mucosal Immunology*, 12(2):457–467, 2019.

[2] A. Muruganandam, F. Migliorini, N. Jeyaraman, et al. Molecular mimicry between gut microbiome and rheumatoid arthritis: Current concepts. *Medical Sciences (Basel)*, 12(4):72, 2024. Published December 12, 2024.

[3] Damian Szklarczyk, Alberto Santos, Christian von Mering, Lars Juhl Jensen, Peer Bork, and Michael Kuhn. Stitch 5: augmenting protein-chemical interaction networks with tissue and affinity data. *Nucleic Acids Research*, 44(D1):D380–D384, 2016.

[4] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, Lars J Jensen, and Christian von Mering. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research*, 47(D1):D607–D613, 2019.

[5] Xiang Wang, Jinjin Sun, Yitao Ding, Zhen Shen, Yu Yang, and Lei Zhang. Deep learning in drug–target interaction prediction: recent advances and future perspectives. *Journal of Cheminformatics*, 16(1):20, 2024.

[6] Rohit Singh, Samuel Sledzieski, Bryan Bryson, Lenore Cowen, and Bonnie Berger. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proceedings of the National Academy of Sciences*, 120(24):e2220778120, 2023.

[7] R. Beccaria, A. Lazzeri, and G. Tiana. Predicting the binding of small molecules to proteins through invariant representation of the molecular structure. *Journal of Chemical Information and Modeling*, 64(17):6758–6767, 2024.

[8] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[9] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29287–29303. Curran Associates, Inc., 2021.

[10] David S Wishart, Eponine Oler, Harrison Peters, AnChi Guo, Sagan Girod, Scott Han, Sukanta Saha, Vicki W Lui, Marcia LeVatte, Vasuk Gautam, Rima Kaddurah-Daouk, and Naama Karu. Mimedb: the human microbial metabolome database. *Nucleic Acids Research*, 51(D1):D611–D620, 10 2022.