Machine Learning-Based Prediction of Functionally Similar Protein Pairs

BIEN 471 – Project Proposal

Liv Toft (260856481, liv.toft@mail.mcgill.ca)

Supervisor
Prof. Brandon Xia, brandon.xia@mcgill.ca

Department of Bioengineering, McGill University
In fulfilment of BIEN 471 – Bioengineering Research Project

April 12, 2024

1 – Abstract

Proteins are the molecular machines of cells, involved in a wide range of tasks from gene regulation to metabolism. Despite millions of protein sequences being known, only a small fraction of their functions have been discovered. This project focuses on developing a computational pipeline for predicting the functions of proteins by identifying functionally similar protein pairs. Various data sources, from protein-protein interaction (PPI) to genomic data, are first combined to capture the profiles of functionally similar and dissimilar protein pairs. Three classification machine learning models – naïve Bayes, multi-layer perceptron (MLP), and XGBoost – were then trained on this data using supervised learning. Both the MLP and XGBoost models had higher sensitivities (0.632 and 0.702) and AUC-ROC scores (0.760 and 0.770) compared to the naïve Bayes model (0.526 sensitivity, 0.714 AUC-ROC). These findings suggest that more complex models with non-linear decision boundaries may be more suitable for this prediction task.

2 – Introduction

2.1 – Background

With the advent of more affordable and accurate sequencing technologies, the number of known protein sequences has exploded since 2011 [1]. However, while there were just under 230 million protein sequence entries in UniProtKB in 2023, only 0.25% of these had functional annotations [2]. This gap can be explained by the slow and costly nature of experimental protein function determination. In order to complete the protein functional map, it is important to use genomic data to infer protein functions from related proteins. Machine learning-based approaches are well suited for this task. They provide a rigorous solution to data integration, allowing for the efficient combination of heterogeneous datasets with differing levels of accuracy and completeness [3]. The use of machine learning to close the gap between known protein sequences and functions has gained huge traction as more powerful algorithms become available and larger datasets are produced.

One common approach is models that predict protein function from sequence [4]. Sequence-based prediction is popular due to the greater availability of protein sequence data compared to other data types, such as protein-protein interaction or protein expression data [4]. While protein function was initially defined by a singular function or role, a more accurate understanding describes a protein as a member of an extensive network of interacting partners [5]. Using machine learning models to capture the relationship between a protein’s function and its role in a protein-protein interaction network has become another approach of interest [4]. These models use computational methods to represent or extract features from protein-protein interaction networks. Yet another approach is the use of structural data. Much of a protein’s function is derived from its 3D structure. Models developed for this data directly extract structure-function relationships from three-dimensional (3D) structural data, unlike feature-based methods [4]. This allows these algorithms to have marked performance over purely sequence-based protein function prediction algorithms [4].

Here, a combination of protein-protein interaction (PPI) network, sequence, and genomic data are used to train machine learning models for identifying functionally similar protein pairs. Understanding the roles of proteins will advance our knowledge of biological systems and processes. This insight will open doors to identifying novel disease genes and therapeutic targets [3, 6]. As such, it is essential to continue to develop accurate and robust machine learning models for the functional annotation of proteins.

2.2 – Machine Learning Models

2.2.1 – Naïve Bayes

Naïve Bayes represents the benchmark model. It is simple to implement and has fast training times. Despite this, because the naïve Bayes model assumes conditional independence between features, it is overly sensitive to correlated features. As a consequence of its simplicity and assumptions about the data, naïve Bayes typically has lower performance compared to more complex machine learning models.

2.2.2 – Multi-Layer Perceptron (MLP)

Previous studies have successfully implemented CNNs, RNNs, GCNs, and other neural networks for protein functional prediction; however, these models utilized sequence, 3D structure, or network data: data where a spatial relationship between features encodes an additional layer of information. DNNs, such as these, perform well on homogenous data – images, audio, and text – where features are more correlated, and the model can train prior knowledge about the inherent structure of the data [7]. The tabular dataset constructed for this project does not have these characteristics. The MLP architecture, where multiple hidden layers are used to apply a series of abstractions to the inputs to map them to the outputs, is better suited for this heterogeneous, unstructured dataset [7, 8].

2.2.3 – XGBoost

XGBoost, short for extreme gradient boosting, is a decision tree ensemble model. This model trains an ensemble of classification and regression trees (CARTs) that individually divide the observations into different groups and assign them specific prediction scores [9]. The final prediction for a given observation is the sum of its prediction scores from each CART [9]. While MLP models have had some success, machine learning models based on decision trees, such as XGBoost, have been shown to outperform DNNs on tabular data [10, 11].

3 – Methods

3.1 – Data

Integrating data from several sources allows for the creation of a set of features that represent the functional similarities of pairs of proteins (Table 1). Data from *S. cerevisiae* was used owing to the availability of larger, more comprehensive data for this model organism. First, physical PPIs were considered because previous studies have found that physically interacting proteins are more likely to share the same function [12]. Here, interactions identified using two different experimental methods, yeast two-hybrid (Y2H) and affinity-capture, are included to mitigate biases and inconsistencies between the datasets. In addition, features summarizing genetic interactions (GIs) were included because pairs of proteins encoded by genes with larger GI scores in magnitude or similar GI profiles are more likely to have similar functions [13]. Gene co-expression was included in the integrated dataset because highly co-expressed genes across various conditions have a higher likelihood of encoding functionally related proteins [14]. Furthermore, co-pathway and co-complex data were selected because proteins that belong to the same biological pathway or the same protein complex often have a similar function. In addition, protein sequence similarity data was used because a high sequence similarity may indicate structural-functional similarity. Finally, sequence similarity was computed as the log(E-value) of the pairwise alignment found using the Basic Local Alignment Search Tool for protein sequences (BLASTp).

SimGIC score is a quantitative measure of a protein pair's functional similarity. It was used as a means to divide each protein pair into one of the two target classes: functionally similar and functionally dissimilar. SimGIC score is calculated using each protein pair's Gene Ontology (GO) annotations and information entropy.

Table 1 – Description of dataset features and sources.

Feature	Experimental Method(s)	Description	Source
Experimental protein-protein interactions (PPI)	Yeast two-hybrid, affinity-capture	Physical protein-protein interactions, experimentally validated by two or more independent sources	BioGRID
Genetic interaction (GI) score	Synthetic genetic array	GI scores for a pair of genes determined by synthetic genetic array	TheCellMap.org [15]
GI profile similarity	Synthetic genetic array	Pearson correlation coefficient for the GI profiles of pairs of genes	TheCellMap.org [15]
PPI structural information	Experimental co-crystal PPI structure and PPI homology models	PPI structural information derived from three-dimensional structures from the Protein Data Bank (PDB) and PPI homology models	Professor Xia's Lab
Co-expression correlation	cDNA microarray hybridization assay	Pearson correlation coefficient for pairs of gene expression profiles corresponding to expression under 300 different mutations and chemical treatments	Hughes et al. [16]
Co-pathway	Various	Whether a pair of proteins belong to the same metabolic pathway	PathBank [17]
Co-complex	Various	Whether a pair of proteins belong to the same protein complex	The Complex Portal [18]
Sequence similarity	BLASTp	Log(E-value) of the pairwise protein sequence alignment using BLASTp	BLAST [19]
Functional similarity score	FastSemSim Python package	SimGIC score calculated using Gene Ontology (GO) annotations and information entropy	GO [20-22]

The histogram of feature values shows that the integrated dataset exhibits a large imbalance in physical interaction, co-pathway, and co-complex (Figure 1). Therefore, the dataset was balanced, giving a total of 13,458 entries.

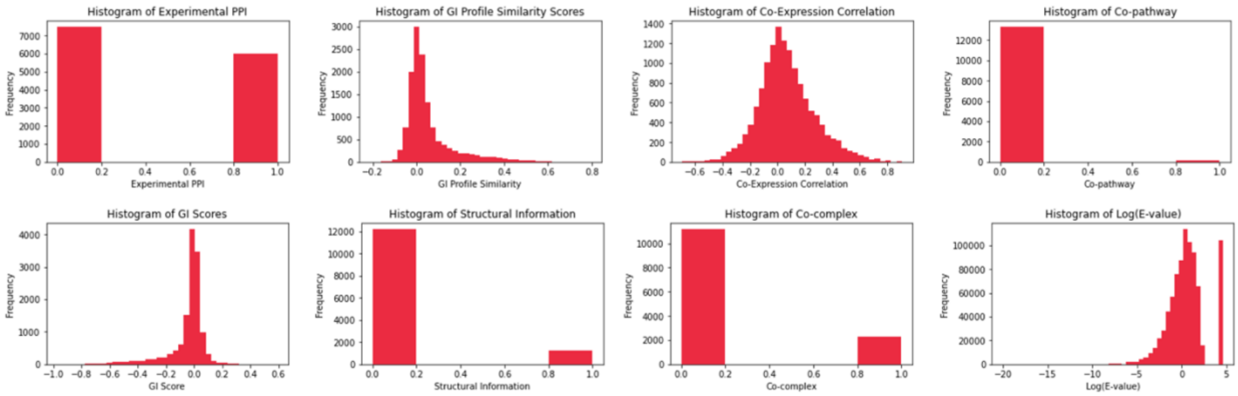


Figure 1 – Value distributions for features in the integrated dataset.

Plotting the correlations between each pair of features shows that no two features are highly correlated, with no Pearson correlation coefficient having a value above 0.55 (Figure 2).

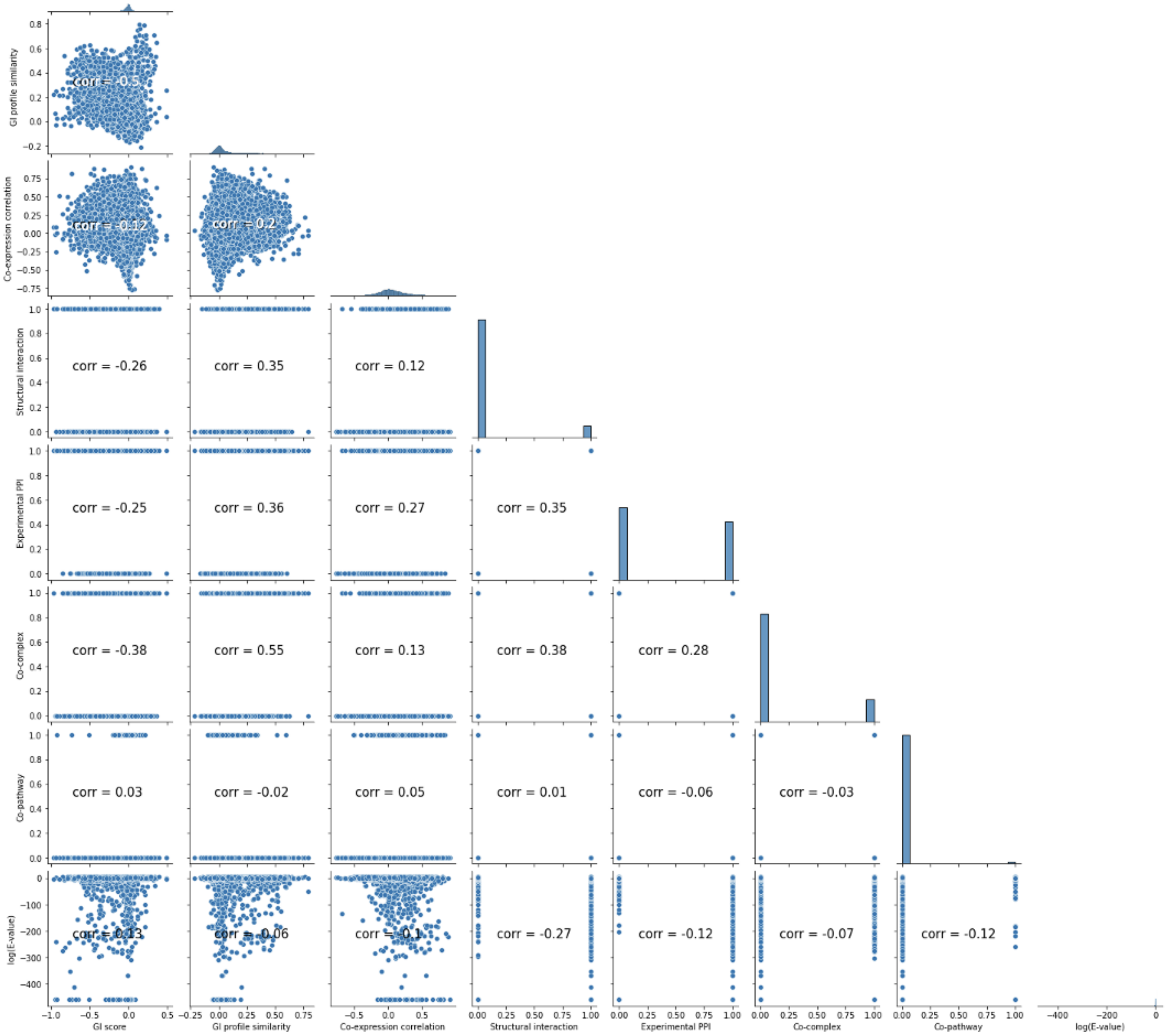


Figure 2 – Pairwise plots for each feature. Diagonal corresponds to value distribution for a singular feature.

The integrated data was divided into a training and testing set according to an 80-20 split. In order to create a classification problem, thresholds for SimGIC score were selected to classify the data into similar and dissimilar protein pairs (Figure 3).

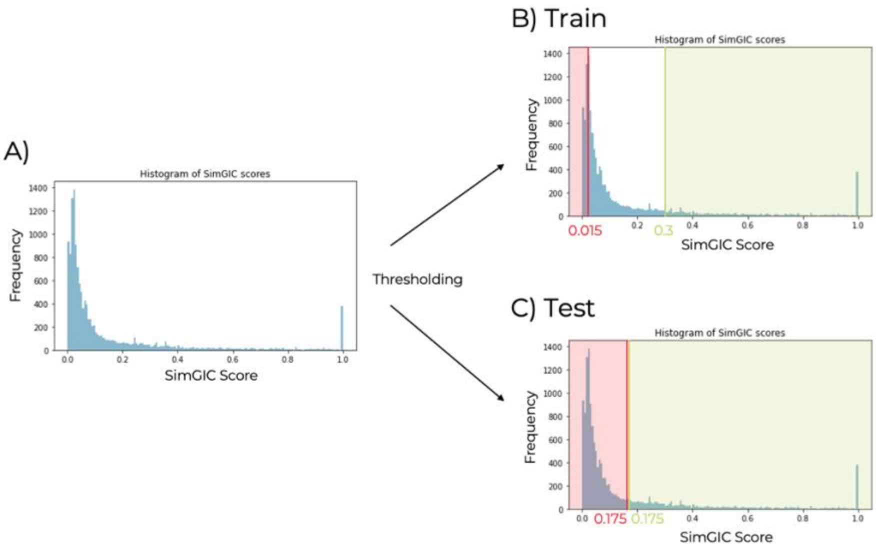


Figure 3 – **A)** Histogram of SimGIC scores. **B)** Thresholds for training set (upper=0.3, lower=0.015). **C)** Thresholds for test set (upper, lower=0.175).

It was observed that there is a large imbalance in the distribution of SimGIC scores, indicating that the majority of protein pairs are dissimilar (Figure 3A). Therefore, most thresholds result in a large class imbalance. In order to mitigate the creation of a model with low sensitivity, upper and lower thresholds were selected such that approximately the same number of similar and dissimilar protein pairs would be included in the training set. SimGIC scores above 0.3 were classified as corresponding to functionally similar protein pairs, while scores below 0.015 were classified as corresponding to functionally dissimilar protein pairs. Protein pairs corresponding to SimGIC scores between the two thresholds were excluded from the training data. This created a balanced training set with 1631 dissimilar and 1542 similar protein pairs. As an additional benefit, this training data only includes highly similar and dissimilar protein pairs. This improved the models' abilities to learn the profiles of similar and dissimilar protein pairs, with performance increasing by 2-20%, depending on the metric (Table 2, Appendix Table 2).

For the test set, the method of upper and lower thresholding was not employed. Instead, a singular threshold was used to annotate the protein pairs. This was done in order to evaluate the model's ability to scale to real-world data. It would be impossible to filter out protein pairs with a SimGIC score between 0.015 and 0.3 without having GO annotations, defeating the purpose of these models as a tool for functionally annotating new proteins with unknown functions. 0.175 was selected as the optimal threshold as it leads to the best sensitivity while preventing a divergence in the other test metrics (Figure 4).

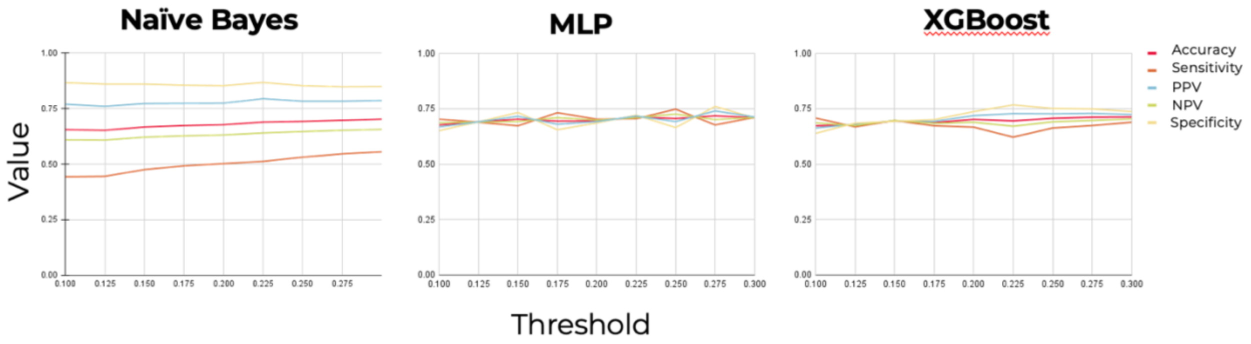


Figure 4 – Model performance metrics for different SimGIC thresholds. 0.175 was selected as the best threshold as it leads to a high sensitivity without sacrificing other metrics.

3.2 – Model Training

The Gaussian naïve Bayes classifier was implemented using the Scikit-learn Python package. The XGBoost classifier was implemented using the XGBoost Python package. Hyperparameter tuning with 10-fold cross-validation was performed using only the training set to identify the optimal number of estimators, maximum model depth, learning rate, and regularization (Appendix Table 1). Finally, the MLP classifier was implemented using the Scikit-learn Python package. Hyperparameter tuning with 10-fold cross-validation was performed using only the training set to identify the optimal number of hidden layers, nodes per hidden layer, alpha, and activation function (Appendix Table 1).

4 – Results & Discussion

The accuracy, sensitivity, positive predictive value, and false positive rate for the three models on the testing dataset are reported in Table 2. These metrics give an idea of each model's ability to correctly identify functionally similar protein pairs (sensitivity, positive predictive value) without over-predicting dissimilar protein pairs (accuracy, false positive rate).

Table 2 – Performance metrics on test set for models trained on double threshold data (0.015, 0.3).

	Accuracy	Sensitivity	Positive Predictive Value	False Positive Rate
Naïve Bayes	0.766	0.526	0.756	0.097
MLP	0.796	0.632	0.763	0.113
XGBoost	0.789	0.702	0.712	0.612

According to these metrics, the MLP and XGBoost models have comparable performance, with the MLP model performing slightly better according to accuracy and positive predictive value. It is also important to note that the models consistently exhibited higher performance metrics when trained on the double-threshold training data compared to the single-threshold training data despite being tested on the same single-threshold test set (Appendix Table 2).

The receiver operating characteristics (ROCs) and associated area under the curves (AUCs) for the three models also indicate that the MLP and XGBoost models are better classifiers (Figure 5). These models have a larger true positive rate to false positive rate ratio at higher cutoff thresholds and, therefore, make better distinctions between the positive and negative classes.

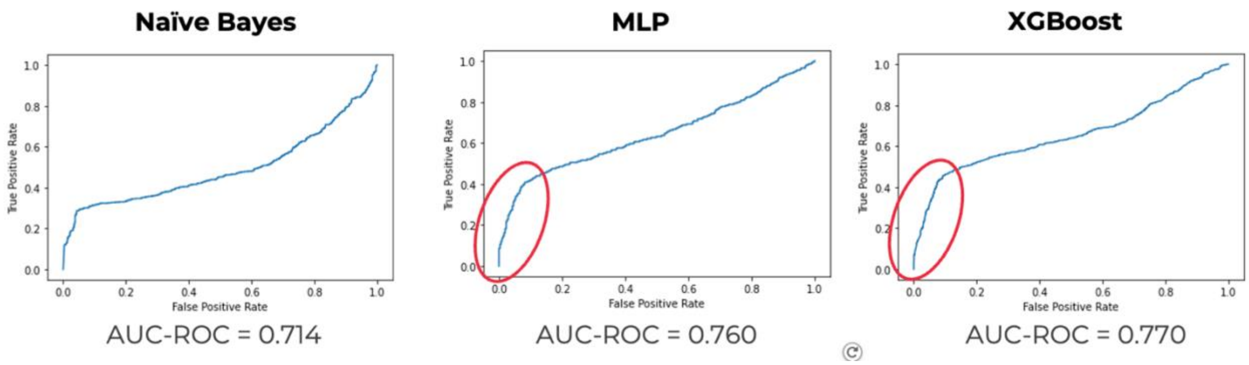


Figure 5 – ROC curves for each model with AUC-ROC scores. Red circles indicate the regions that show why the MLP and XGBoost models perform better than naïve Bayes.

Overall, the MLP and XGBoost models consistently performed better than the naïve Bayes model across most metrics. One key distinction between the models is that the naïve Bayes classifier uses a linear decision boundary, while the other two models do not. Therefore, the better performance of these models many indicate that a non-linear decision boundary is more suitable for this classification problem.

5 – Future Work

Future work may investigate the optimization of the integrated dataset. For example, during GO annotation, a protein may be given the same annotation as another protein if the two proteins have a PPI reported in BioGRID. BioGRID PPIs were also a feature used to predict functionally similar proteins. However, functionally similar protein pairs were determined using their SimGIC score, which is based on GO annotations. This introduces circular reasoning. Here, it was assumed that there are few enough of these instances that model performance would remain the same even after removing proteins with GO annotations that are based on BioGRID PPIs. However, for verification, these proteins should be removed and the model performances re-evaluated. Furthermore, more data can be added to the integrated dataset. Such data may include protein expression correlation across different cell lines, network data such as the Jaccard index, and Pfam domain data.

Framing the problem as a binary classification problem posed difficulties for the machine learning models as they struggled to classify pairs of proteins corresponding to 'ambiguous' SimGIC scores. Implementing regression or multiclass classification models to directly predict SimGIC scores or a wider range of protein pair similarity types (E.g., very similar, medium similarity, neutral, etc.) may address this issue. Once a high-performing model is achieved, it can be used to construct an accurate functional linkage network that could then be used to investigate and identify disease genes and drug targets.

6 – Conclusion

Here, multiple datasets from different sources were combined to create an integrated dataset with the goal of capturing the unique profiles of functionally similar and dissimilar pairs of proteins. The dataset was used to train three machine learning models – naïve Bayes, MLP, and XGBoost – for predicting functionally similar protein pairs. SimGIC score was used as a quantitative measurement of protein pair similarity. Two SimGIC score thresholds (upper=0.3, lower=0.175) were used to classify similar and dissimilar protein pairs in the training dataset, while a singular threshold (upper, lower=0.175) was used for the test set. Using two thresholds for the training set balanced the two classes and allowed the models to only train on examples of strongly similar or dissimilar protein pairs, exhibiting better performance compared to models trained on single-threshold training data (threshold=0.175) across most metrics. Testing on un-balanced, single-threshold, 'real-world' data demonstrates how model performance holds despite being trained on data with a different class distribution.

Overall, the MLP and XGBoost models exhibited improved performance over the naïve Bayes model according to most performance metrics. These models also make "stronger" predictions (i.e., assign higher probabilities) to positively predicted protein pairs. This improved performance may suggest that a non-linear decision boundary is more suitable for this problem.

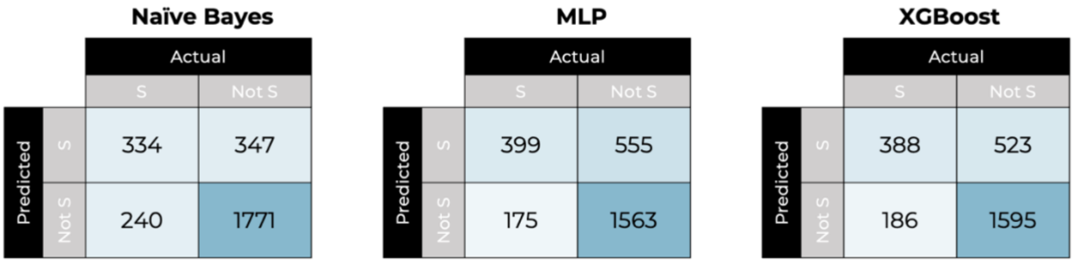
These models are promising first steps towards creating a pipeline for protein functional annotation. With an optimized model, work towards predicting the functions of the 99.75% of unannotated protein sequences can begin. This would open doors in broadening our understanding of biological systems and processes, aid in the discovery of novel disease genes, and potentially lead to the identification of new drug targets.

7 – Appendix

Table 1 – Optimal hyperparameters.

	MLP
Number of hidden layers	2
Nodes per layer	20, 60
Activation function	Rectifier linear unit
Alpha	5e-6

	XGBoost
Learning rate	0.12
Depth	4
Number of estimators	80
Lambda regularization	3



Naïve Bayes

		Actual	
		S	Not S
Predicted	S	334	347
	Not S	240	1771

MLP

		Actual	
		S	Not S
Predicted	S	399	555
	Not S	175	1563

XGBoost

		Actual	
		S	Not S
Predicted	S	388	523
	Not S	186	1595

Figure 1 – Confusion matrices for the test set (threshold=0.175) for models trained on the double-threshold training set (thresholds=0.015,0.3)

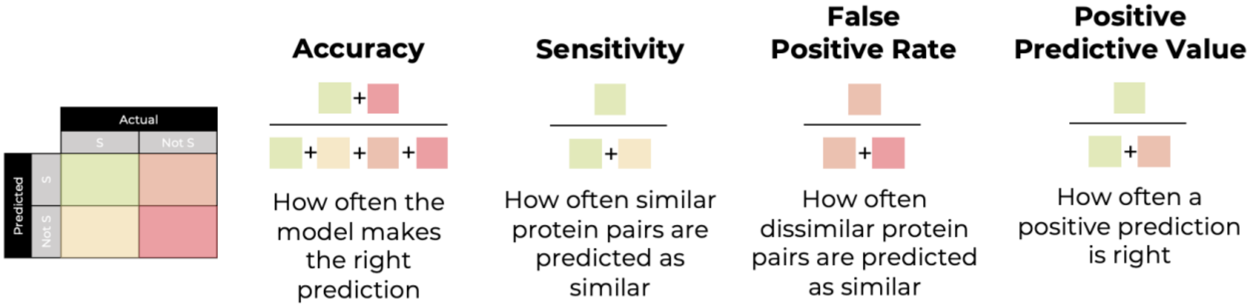

Actual
S
Not S
Predicted
S
Not S
Accuracy
How often the model makes the right prediction
Sensitivity
How often similar protein pairs are predicted as similar
False Positive Rate
How often dissimilar protein pairs are predicted as similar
Positive Predictive Value
How often a positive prediction is right

Figure 2 – Performance metrics calculations.

Table 2 – Performance metrics on test set for models trained on single threshold data (0.175).

	Accuracy	Sensitivity	Positive Predictive Value	False Positive Rate
Naïve Bayes	0.785	0.396	0.521	0.104
MLP	0.805	0.313	0.628	0.053
XGBoost	0.751	0.658	0.459	0.222

References

[1] M. Varadi *et al.*, "3D-Beacons: decreasing the gap between protein sequences and structures through a federated network of protein structure data resources," *GigaScience*, vol. 11, 2022, doi: https://doi.org/10.1093/gigascience/giac118.

[2] L. A. Bugnon, E. Fenoy, A. A. Edera, J. Raad, G. Stegmayer, and D. H. Milone, "Transfer learning: The key to functionally annotate the protein universe," *Patterns*, vol. 4, no. 2, 2023, doi: 10.1016/j.patter.2023.100691.

[3] B. Linghu, E. A. Franzosa, and Y. Xia, "Construction of functional linkage gene networks by data integration," *Data Mining for Systems Biology*, vol. 939, pp. 215-232, 2013, doi: https://doi.org/10.1007/978-1-62703-107-3_14.

[4] T.-C. Yan *et al.*, "A systematic review of state-of-the-art strategies for machine learning-based protein function prediction," *Computers in Biology and Medicine*, vol. 154, 2023, doi: https://doi.org/10.1016/j.compbiomed.2022.106446.

[5] R. Bonetta and G. Valentino, "Machine learning techniques for protein function prediction," *Proteins*, vol. 88, no. 3, pp. 397-413, 2019, doi: https://doi.org/10.1002/prot.25832.

[6] A. Derry and R. B. Altman, "Explainable protein function annotation using local structure embeddings," *bioRxiv*, 2023, doi: https://doi.org/10.1101/2023.10.13.562298.

[7] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022, doi: 10.1109/TNNLS.2022.3229161.

[8] J. Brownlee. "Crash course on multi-layer perceptron neural networks." Machine Learning Mastery. https://machinelearningmastery.com/neural-networks-crash-course/ (accessed January 28, 2025).

[9] XGBoost developers. "Introduction to Boosted Trees." XGBoost Tutorials. https://xgboost.readthedocs.io/en/stable/tutorials/model.html (accessed January 28, 2024).

[10] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," presented at the 36th Conference on Neural Information Processing Systems, New Orleans, USA, 2022.

[11] R. Shwatrz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84-90, 2022, doi: https://doi.org/10.1016/j.inffus.2021.11.011.

[12] P. M. Kim, L. J. Lu, and M. B. Gerstein, "Relating three-dimensional structures to protein networks provides evolutionary insights," *Science*, vol. 314, no. 5807, pp. 1938-1941, 2006, doi: 10.1126/science.1136174.

[13] M. Costanzo *et al.*, "The genetic landscape of a cell," *Science*, vol. 327, no. 5964, pp. 425-431, 2010, doi: 10.1126/science.1180823.

[14] A. Grigoriev, "A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage T7 and the yeast *Saccharomyces cerevisiae*," *Nucleic Acids Research*, vol. 29, no. 17, pp. 3513-3519, 2001, doi: 10.1093/nar/29.17.3513.

[15] M. Usaj *et al.*, "TheCellMap.org: A web-accessible database for visualizing and mining the global yeast genetic interaction network," *G3: Genes, Genomes, Genetics*, vol. 7, no. 5, pp. 1539-1549, 2017, doi: 10.1534/g3.117.040220.

[16] T. R. Hughes *et al.*, "Functional discovery via a compendium of expression profiles," *The Cell*, vol. 102, no. 1, pp. 109-126, 2000, doi: https://doi.org/10.1016/S0092-8674(00)00015-5.

[17] D. S. Wishart *et al.*, "PathBank: a comprehensive pathway database for model organisms," *Nucleic Acids Research*, vol. 48, no. D1, pp. D470-D478, 2020, doi: 10.1093/nar/gkz861.

[18] B. H. M. BMeldal, H. Bye-A-Jee, F. Melicher, and L. Perfetto, "Complex Portal 2018: extended content and enhanced visualization tools for macromolecular complexes," *Nucleic Acids Research*, vol. 47, no. D1, pp. D550-D558, 2019, doi: https://doi.org/10.1093/nar/gky1001.

[19] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403-410, 1990, doi: 10.1016/S0022-2836(05)80360-2.

[20] M. K. Cherry *et al.*, "Saccharomyces Genome Database: the genomics resource of budding yeast," *Nucleic Acids Research*, vol. 40, pp. D700-D705, 2012, doi: 10.1093/nar/gkr1029.

[21] M. Ashburner *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25-29, 2000, doi: 10.1038/75556.

[22] The Gene Ontology Consortium, "The Gene Ontology knowledgebase in 2023 " *Genetics*, vol. 224, no. 1, 2023, doi: 10.1093/genetics/iyad031.