AI BioInnovative Challenge

Protein Engineering

Problem Statement – 3 (PE3)

A model for predicting moon light proteins based on protein language models and deep learning.

Submitted by

Team: InnoVisonaries

Members:
Arushi Gupta
Anik Roy
Somil Gupta

1.Understanding the Problem Statement

1.1 Problem Statement

PE3- A model for predicting moon light proteins based on protein language models and deep learning.
The primary objective of this problem statement is to design and implement a predictive model utilizing protein language models and deep learning techniques. The model's purpose is to discern patterns and features from the given protein sequences in order to accurately predict whether a protein is likely to be a moonlight protein.

1.2 Background

The identification of moonlight proteins, characterized by their ability to perform multiple functions or localize in diverse cellular compartments, is a critical aspect of proteomic research. These proteins exhibit behaviors beyond the conventional expectations associated with their primary roles, making their accurate prediction a challenging yet essential task.

1.3 Significance of Moonlight Proteins

Moonlight proteins play a pivotal role in cellular processes and are implicated in various biological phenomena. Understanding their functions and predicting their occurrence can provide valuable insights into the complex landscape of cellular activities.

1.4 Data Representation

A dataset of 351 proteins was provided that contained 136 non-moonlighting and 215 moonlighting proteins. This dataset contained proteins derived from different organisms. The dataset, formatted in FASTA files, serves as the foundation for our project. FASTA is a widely used and simple text-based format for representing nucleotide or protein sequences. The data is categorized into two main groups:

Positive Files: Containing sequences of proteins identified as moonlight proteins, the positive files represent instances where a protein exhibits multifunctionality or unexpected localizations within the cell.

Negative Files: Comprising sequences of proteins deemed as non-moonlight proteins, the negative files act as a contrast to the positive dataset, encompassing proteins that adhere to conventional functional expectations.

2. Data Handling and Analysis

2.1 Feature Extraction

- Feature Extraction using ftrCool Tool: Our feature extraction process involved the utilization of the ftrCool tool, a powerful resource in the R language specifically designed for extracting informative features from biological sequences. We extracted the Amino Acid k part feature. Using the same tool, we stored our features for both the files in two separate csv files.

- Combining both the dataset: The datasets were merged based on a common identifier, such as protein sequence ID, to ensure accurate alignment of features. The resultant dataset, named combined_data.csv, represents a consolidated feature set that integrates both moonlight proteins and non-moonlight protein features for each protein sequence.

- Labelling: Proteins in positive files were labeled as moonlight proteins (class 1), while those in negative files were labeled as non-moonlight proteins (class 0). This labeling step is crucial for supervised learning, allowing the model to associate features with the respective classes during training.

2.2 Data Preprocessing

- Oversampling with SMOTE: Given the limited size of the dataset, we addressed class imbalance using Synthetic Minority Over-sampling Technique (SMOTE). This technique artificially increased the representation of the minority class (moonlight proteins), enabling a more balanced distribution for model training.

- Manual Data Augmentation: Data augmentation is a crucial technique in machine learning, particularly when dealing with limited datasets. In the context of moonlight protein prediction, where the available data may be scarce, manual data augmentation becomes an essential strategy to enhance model generalization and performance.

- Feature Scaling with StandardScaler: To standardize feature scales and facilitate optimal model convergence, we employed StandardScaler. This preprocessing step ensured that all features contributed equally to the model training process, avoiding biases induced by differing scales.

- Data splitting: The dataset was divided into training and validation sets to facilitate model training and performance evaluation. A standard split, such as 80% for training and 20% for validation, was employed to ensure an unbiased assessment of the model's generalization capabilities.

2.3 Exploratory Data Analysis

- Sequence Length Distribution: Analyzing the distribution of protein sequence lengths allowed us to make informed decisions about model architecture, such as setting appropriate input dimensions.

- Class Distribution: Understanding the balance between positive and negative samples helped us address potential biases and implement strategies to handle class imbalance during training.

3. Methodology

3.1 Model Selection

In the pursuit of effective moonlight protein prediction, a thoughtful selection of machine learning models was undertaken, encompassing a variety of methodologies to address the intricacies of the task. Initially, we considered deep learning models due to their ability to capture complex patterns, but encountered challenges related to the limited size of our dataset, which adversely affected the accuracy of these models. Consequently, we opted for a diverse set of traditional machine learning models, each renowned for its distinctive characteristics:

- **Support Vector Machine (SVM):** Selected for its capability to handle intricate decision boundaries, SVM plays a crucial role in identifying hot spots within proteins.

- **K-Nearest Neighbors (KNN):** Leveraging proximity-based classification, KNN is employed for predicting hot spots in proteins and exploring Protein-Protein Interactions (PPI).

- **Naive Bayes (NB):** Recognized for assuming independence among features, NB is chosen for its simplicity and efficiency, with successful applications reported in the context of PPI.

- **Decision Tree (DT):** Constructing hierarchical decision-making structures, DT is successful in PPI prediction and has demonstrated favorable outcomes in the specific domain of human virus PPI.

- **Random Forest (RF):** Harnessing the power of ensemble learning through multiple decision trees, RF has proven efficacy in predicting PPI.

- **Multi-Layer Perceptron (MLP):** Employing a neural network architecture for learning complex patterns, MLP has shown promising results, particularly in the prediction of human virus PPI.

- **AdaBoost (ADA):** Integrating weak learners into a robust ensemble, ADA aims to minimize classifier errors, contributing to enhanced prediction accuracy.

- **Logistic Regression (LR):** As a linear model suitable for binary classification, LR has demonstrated success in predicting protein function from protein-protein interaction data.

3.2 Stratified K-Fold Cross Validation

To evaluate model performance robustly, we adopted Stratified K-Fold Cross-Validation with 100 folds. This approach ensures that each fold maintains the same class distribution as the original dataset, preventing bias in model evaluation

3.3 Model Training and Evaluation

For each model in the ensemble, we performed cross-validation to assess its accuracy across multiple folds. Additionally, the models were individually trained on the entire resampled dataset and evaluated on a separate test set.

3.4 Ensemble Averaging

A unique aspect of our methodology lies in the utilization of ensemble averaging. Instead of selecting a single model for prediction, we averaged the predictions from multiple models. This approach leverages the diversity of individual models, potentially improving overall predictive performance.

3.5 Results and Evaluation

The performance of each model was evaluated based on cross-validation accuracy, and the ensemble's effectiveness was measured by averaging the predictions. The final ensemble accuracy serves as a comprehensive metric, reflecting the combined strength of multiple models.

3.6 Uniqueness of the Method

Several key aspects contribute to the uniqueness of the method:

- Diverse Model Selection: The inclusion of a diverse set of machine learning models, ranging from traditional algorithms like SVM and decision trees to more complex models like neural networks, ensures a thorough exploration of different learning paradigms. This adaptability allows the methodology to capture intricate patterns in the data that might be missed by a single model.

- Stratified K-Fold Cross-Validation: The use of Stratified K-Fold Cross-Validation with a high number of folds (100) ensures robust evaluation. By maintaining the class distribution in each fold, the methodology provides a more accurate representation of the models' generalization capabilities on imbalanced datasets, which is a crucial consideration for moonlight protein prediction.

- Oversampling with SMOTE: The application of Synthetic Minority Over-sampling Technique (SMOTE) directly addresses the issue of class imbalance in the dataset. This technique generates synthetic instances of the minority class, allowing the models to learn more effectively from positive instances without introducing biases.

- Ensemble Averaging: The introduction of ensemble averaging is a unique and adaptive approach to making predictions. Instead of relying on a single model, the methodology leverages the strengths of multiple models through averaging. This ensemble technique aims to mitigate individual model biases and enhance overall prediction accuracy.

- Suppression of Convergence Warnings: The methodology incorporates measures to suppress ConvergenceWarnings, which can be particularly relevant when dealing with complex models or datasets. This demonstrates a meticulous attention to the stability and efficiency of the training process.

The methodology for moonlight protein prediction integrates a thoughtful selection of models, addresses class imbalance through oversampling, employs cross-validation for robust evaluation, and introduces ensemble averaging for enhanced predictive capabilities. These elements collectively contribute to the uniqueness and adaptability of the method for the specific challenges posed by the problem statement.

4. Mathematical and Logical Reasoning

4.1 Oversampling with SMOTE

- Mathematical Reasoning: Class Imbalance Mitigation: The Synthetic Minority Over-sampling Technique (SMOTE) introduces synthetic instances to the minority class, mathematically balancing the class distribution. The oversampling process involves creating synthetic examples along line segments connecting minority class instances, thereby enhancing the representation of moonlight proteins.
- Logical Reasoning: Improved Generalization: By increasing the representation of moonlight proteins through oversampling, the logical inference is that the models will generalize better to unseen instances of these proteins. This aligns with the expectation that a more balanced dataset contributes to more robust and accurate predictions.

4.2 Stratified K-Fold Cross-Validation

- Mathematical Reasoning: Stratification for Class Balance: The use of Stratified K-Fold Cross-Validation mathematically ensures that each fold maintains a proportionate representation of moonlight proteins and non-moonlight proteins. This stratification is achieved by preserving the class distribution in each fold, preventing biases during model evaluation.
- Logical Reasoning: Robust Evaluation: Strategically chosen cross-validation ensures that the models are rigorously evaluated across various subsets of the data. The logical reasoning is that this approach provides a more reliable

assessment of model performance, particularly in the context of imbalanced datasets.

4.3 Ensemble Averaging

- Mathematical Reasoning
  Averaging Predictions: Ensemble averaging involves summing predictions from individual models and dividing by the number of models. Mathematically, this process creates an amalgamation of predictions, effectively smoothing out variations and potentially reducing prediction errors.
- Logical Reasoning: Robust Evaluation: Diverse Model Contributions: The logical inference is that the diversity of models contributes different perspectives and patterns. Averaging predictions provides a consolidated and potentially more accurate prediction by mitigating biases associated with individual models.

4.4 Model Training and Evaluation

- Mathematical Reasoning: Optimization with Training Data: During model training, mathematical optimization techniques are employed to adjust model parameters iteratively. The logical reasoning is that this process optimizes the model to discern patterns and relationships within the training data.
- Logical Reasoning: Robust Evaluation: Evaluation Metrics: The choice of accuracy as an evaluation metric aligns with the logical reasoning that accurately predicting both moonlight and non-moonlight proteins is crucial. This reflects the overall effectiveness of the models in making correct predictions.

5. Evaluation Metrics and Result Analysis

5.1 Cross-Validation Results

- SVM (Support Vector Machine): 0.987
- KNN (K-Nearest Neighbors): 0.992
- NB (Naive Bayes): 0.625
- DT (Decision Tree): 0.9911
- RF (Random Forest): 0.9913
- MLP (Multi-Layer Perceptron): 0.9921
- ADA (AdaBoost): 0.9241
- LR (Logistic Regression): 0.8973

5.2 Insights

- Comparative Analysis: The ensemble approach demonstrates significant improvement over individual models, particularly outperforming in accuracy.
- Identification of Strengths and Weaknesses: Decision Tree, Random Forest, and Multi-Layer Perceptron exhibit perfect accuracy on the test set, showcasing their strength in moonlight protein prediction.
- Insights into Moonlight Protein Prediction: The high accuracy of ensemble averaging implies a robust and reliable prediction mechanism for moonlight proteins.
- Potential Areas for Improvement: Naive Bayes shows lower performance, suggesting potential areas for refinement in future iterations.

6. Future Scope

As we navigate the current landscape of moonlight protein prediction, several avenues for future exploration and enhancement emerge.

1. Incorporation of Advanced Feature Engineering:
    - Explore and incorporate advanced feature engineering techniques that leverage the latest advancements in bioinformatics and protein sequence analysis.
    - Consider domain-specific features or representations that may enhance the discriminative power of the models.

2. Integration of Deep Learning Architectures:
    - Investigate the applicability of more sophisticated deep learning architectures, such as recurrent neural networks (RNNs) or attention mechanisms.
    - These architectures may capture long-range dependencies and intricate patterns present in protein sequences.

3. Utilization of Graph Neural Networks (GNNs):
    - Explore the application of Graph Neural Networks (GNNs) to model the complex relationships and interactions within protein sequences.
    - GNNs have shown promise in capturing structural information and dependencies in biological data.

4. Transfer Learning in Bioinformatics:

- Explore the potential of transfer learning techniques in bioinformatics, where pre-trained models on large protein databases can be fine-tuned for moonlight protein prediction tasks.
- Transfer learning may enhance model generalization and efficiency.

5. Integration of Multi-Omics Data:
    - Extend the analysis to incorporate multi-omics data, integrating information from various biological sources (genomics, transcriptomics, proteomics).
    - The inclusion of multi-modal data may provide a more comprehensive understanding of moonlight protein behavior.

In conclusion, the future scope of moonlight protein prediction is expansive and dynamic, offering opportunities for interdisciplinary collaboration, technological innovation, and a deeper understanding of the functional dynamics of proteins

References

1. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04194-5#Bib1
2. https://academic.oup.com/bioinformatics/article/38/8/2102/6502274
3. https://cran.r-project.org/web/packages/ftrCOOL/ftrCOOL.pdf