Alma Mater University of Bologna
Bioinformatics master's degree - 2023
Laboratory of Bioinformatics 2 Project Report

Comparative Analysis of Machine Learning Methods for Signal Peptide Detection

Nafiseh Barmakhshad

Department of Pharmacy and Biotechnology, FaBiT

Abstract

Signal peptides play a crucial role in directing proteins to their proper cellular and extracellular locations, making their accurate detection essential for understanding protein function and localization. This review article compares various machine learning methods used for signal peptide detection, including the von Heijne method, Support Vector Machines (SVM), SignalP, DeepSig, and Fully Connected Neural Networks (FCNN).

The von Heijne method, a rule-based approach, demonstrates high reliability and precision with an accuracy of 99.54% on the test set and 99.63% on the benchmarking set. SVM and SignalP models also show exceptional performance, achieving accuracy levels of 99.29% and 98.86% respectively, leveraging the strengths of machine learning algorithms to detect complex sequence patterns. DeepSig, employing deep convolutional neural networks, offers significant improvements over traditional methods but exhibits lower recall, indicating potential areas for enhancement. The FCNN model, despite various optimization attempts, did not surpass the performance of a random classifier, highlighting the need for more sophisticated architectures or refined feature representations.

Our comprehensive evaluation of these models underscores the effectiveness of traditional and machine learning approaches in signal peptide detection, while also identifying the limitations and future directions for improvement. This study emphasizes the potential of machine learning methods to advance the field of bioinformatics, enhancing our understanding of protein sorting and compartmentalization.

Introduction

Protein sorting and compartmentalization are intricate biological mechanisms essential for proper cellular function. Signal peptides are short sequence segments located at the N-termini of newly synthesized proteins, guiding them towards the secretory pathway (von Heijne, 1990). These peptides direct proteins to various cellular and extracellular locations, such as the endoplasmic reticulum, Golgi apparatus, and plasma membrane, and are crucial for the translocation of proteins across the cytoplasmic membrane via the well-established Sec pathway in both eukaryotic and prokaryotic cells (Martoglio and Dobberstein, 1998). Typically, signal peptides are organized into three distinct domains: the positively charged N-region, the central hydrophobic H-region, and the polar uncharged C-region containing the cleavage site. The accurate prediction and identification of these signal peptides are pivotal for understanding protein function and destination.

With the avalanche of protein sequences in the post-genomic era, timely utilization of their information is essential to stimulate the development of medical science and expedite the course of drug design. Many new techniques in computational biology have been developed to address this need. However, to effectively use the knowledge of signal peptides, one must first identify the signal peptides and their cleavage sites. Signal peptides direct proteins to their proper cellular and extracellular locations. One major example of such a process is the translocation of proteins across the cytoplasmic membrane via the well-established Sec pathway found in both eukaryotic and prokaryotic cells. In this secretory pathway, proteins designated for export from the cell are labeled by an N-terminal signal sequence. This signal sequence directs its protein to the secretion apparatus. After translocation of the protein across the cell membrane, the N-terminal signal peptide is usually cleaved off by an extracellular signal peptidase.

Signal peptides are always organized in three distinct domains. The polar C-domain (5 to 7 amino acids long) is often of the type Ala-X-Ala. The central H-domain contains essentially Val, Ile, Leu, Ala, Phe, Met, and Trp residues and has a mean length of 12 amino acids. The high hydrophobicity of this domain seems to be essential for the function of a signal sequence. The N-terminal domain (1 to 2 amino acids long) contains Lys or Arg residues.

Accurately detecting signal peptides in protein sequences is critical in bioinformatics and proteomics. It enables researchers to predict the subcellular localization of proteins, which is essential for elucidating protein function, understanding disease mechanisms, and designing therapeutic interventions. The knowledge of signal peptides facilitates the identification of secreted proteins and membrane-bound proteins, playing a vital role in drug design and the development of medical science. The detection of signal peptides and their cleavage sites can also aid in annotating newly sequenced genomes, thus enhancing our understanding of the proteome (Choo et al., 2009).

The primary objective of this article is to compare various machine learning methods for detecting signal peptides in protein sequences. The methods under comparison include the von Heijne method, Support Vector Machine (SVM), SignalP, DeepSig, and a fully connected neural network. This comparison will be based on their performance after training, assessed on test and benchmarking datasets. By evaluating these models, we aim to determine the most effective approach for signal peptide detection and cleavage site prediction, thereby contributing to the advancement of computational methods in bioinformatics. Several computational methods have been developed to detect signal peptides, leveraging machine learning models trained on available experimental data. Early methods like the von Heijne approach utilized weight matrices for prediction (von Heijne, 1983). More recent and sophisticated methods have employed machine learning techniques such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and Hidden Markov Models (HMMs) to improve prediction accuracy (Nugent and Jones, 2009; Petersen et al., 2011).

Positive Dataset consists of eukaryotic sequences with experimentally determined signal peptides (SPs) yielded 2969 sequences. The data were filtered to include only sequences with reviewed protein annotation status and a minimum length of 30 residues. Search Input: (taxonomy_id:2759) AND (reviewed:true) NOT (length:[0 TO 30]) AND (ft_signal_exp:*)

Negative Dataset includes eukaryotic sequences without SPs returned approximately 25,000 sequences.These sequences are annotated with experimental evidence for localization in cellular compartments unrelated to SPs (e.g., cytosol, nucleus, mitochondrion, plastid, peroxisome, cell membrane), with a minimum length of 30 residues. Search Input: (taxonomy_id:2759) AND (reviewed:true) AND (cc_scl_term_exp:"cytosol, nucleus, mitochondrion, plastid, peroxisome, cell membrane") NOT (ft_signal:*) NOT (length:[0 TO 30])

To ensure non-redundancy, clustering was performed using mmseq2 with a maximum pairwise sequence similarity of 30% and alignment coverage of 40%. After clustering, representative proteins were split into training (80%) and benchmarking (20%) sets, maintaining a 1:10 ratio of positive to negative sequences. Negative and positive datasets were independently shuffled and divided into two subsets: 80% for training and 20% for benchmarking.

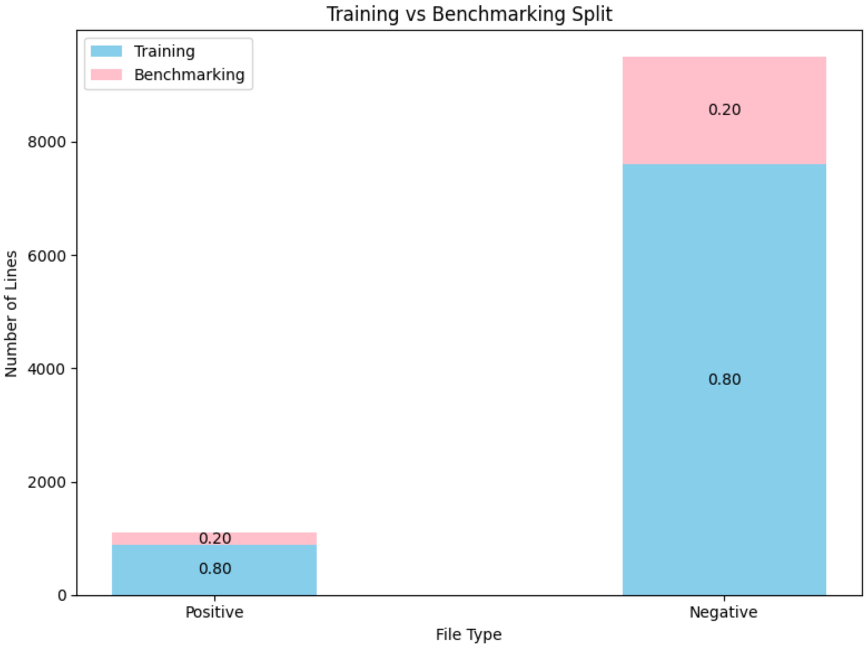


Figure 1: Training vs Benchmarking Split

Statistical Analysis and Visualization

The comprehensive preparation and analysis of the datasets, including ensuring no overlap between training and benchmarking sets, analyzing signal peptide lengths, and comparing amino acid compositions, establish a solid foundation for evaluating machine learning models for signal peptide detection. The visualizations confirm that the datasets are balanced and representative, providing confidence in the robustness of subsequent model evaluations.

SP Length Distribution:

The histogram depicts the distribution of signal peptide (SP) lengths in the training and benchmarking datasets. Both datasets show a similar distribution, with most SP lengths concentrated between 15 and 30 residues, indicating a representative sample for model training and benchmarking.

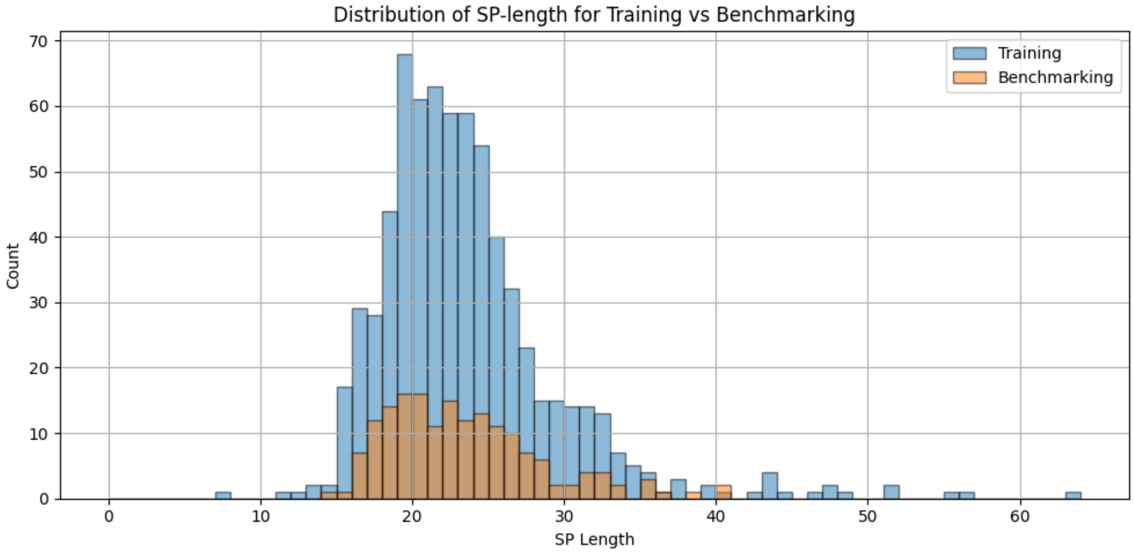


Figure 2: Distribution of SP-length for Training vs Benchmarking

Density Plot Analysis:

The density plots illustrate the distribution of sequence lengths for positive and negative datasets. The left plot shows the training data, while the right plot shows the benchmarking data. Positive sequences are represented in red, and negative sequences in blue. Both plots exhibit overlapping distributions, indicating consistency and balance between the training and benchmarking sets, which is crucial for effective model evaluation.

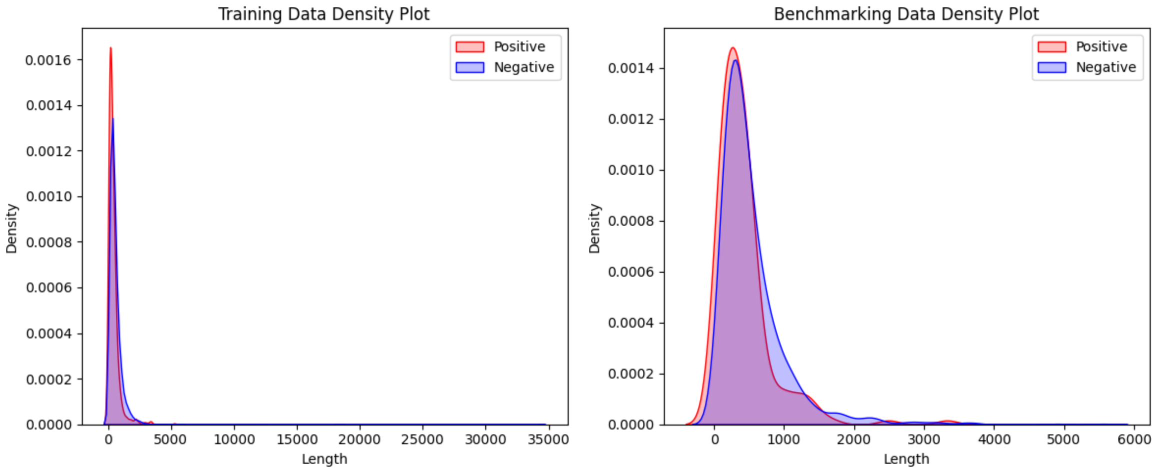


Figure 3: Training and Benchmarking Data Density Plot

Taxonomic Kingdom Distribution:

Pie charts displaying the distribution of taxonomic kingdoms within the training and benchmarking datasets. The left chart represents the training data, and the right chart represents the benchmarking data. The distributions are divided into Metazoa (blue), Viridiplantae (green), Fungi (orange), and Others (red). Both charts show a balanced distribution among different taxonomic kingdoms, ensuring diversity and representativeness in both training and benchmarking sets, which is essential for robust model evaluation

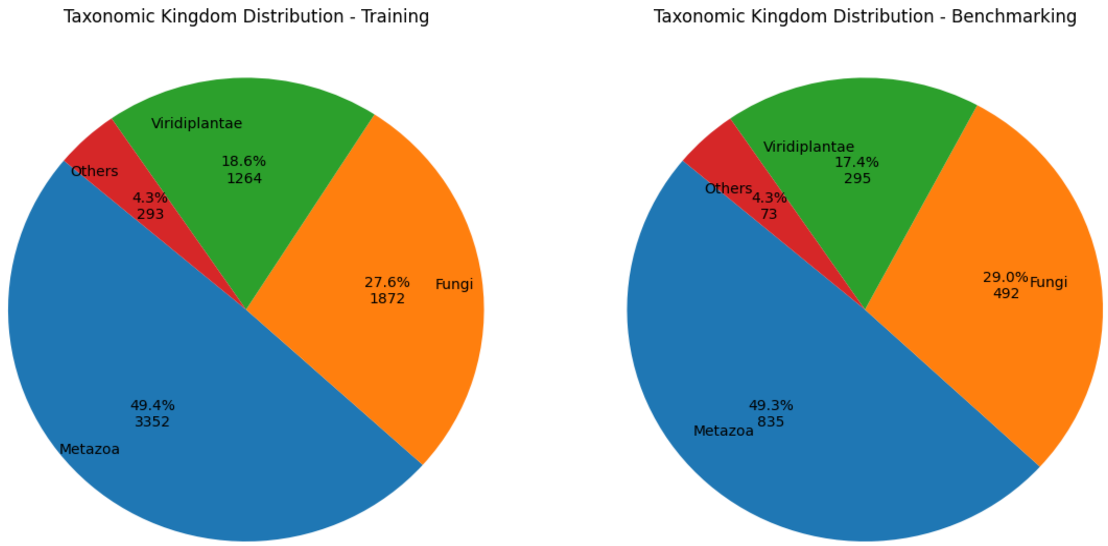


Figure 4: Taxonomic Kingdom Distribution in Training and Benchmarking Sets

Amino Acid Composition Comparison:

Bar chart comparing the amino acid compositions of the training dataset (blue), benchmarking dataset (orange), and the SwissProt background distribution (green). Each bar represents the percentage of a specific amino acid (listed along the x-axis) in the respective datasets. This comparison illustrates the relative frequencies of each amino acid, highlighting the higher presence of Alanine (A) and Leucine (L) in signal peptides. The balanced representation of amino acids across training and benchmarking datasets, compared to the SwissProt background, validates the quality and representativeness of the datasets used for model training and evaluation.

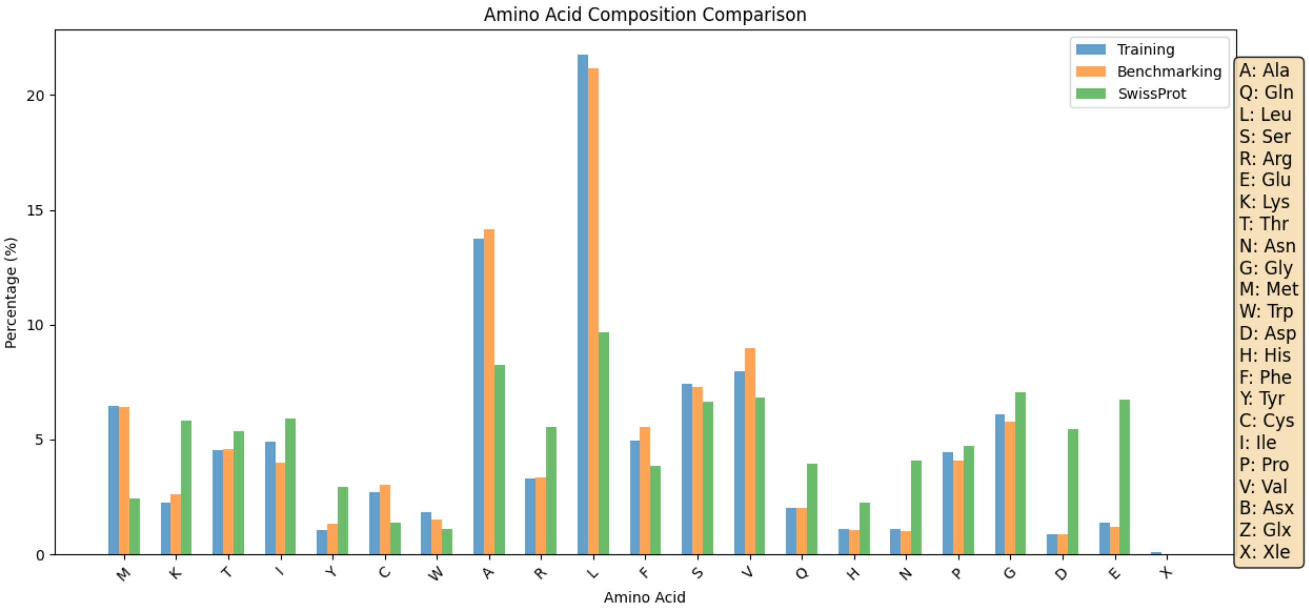
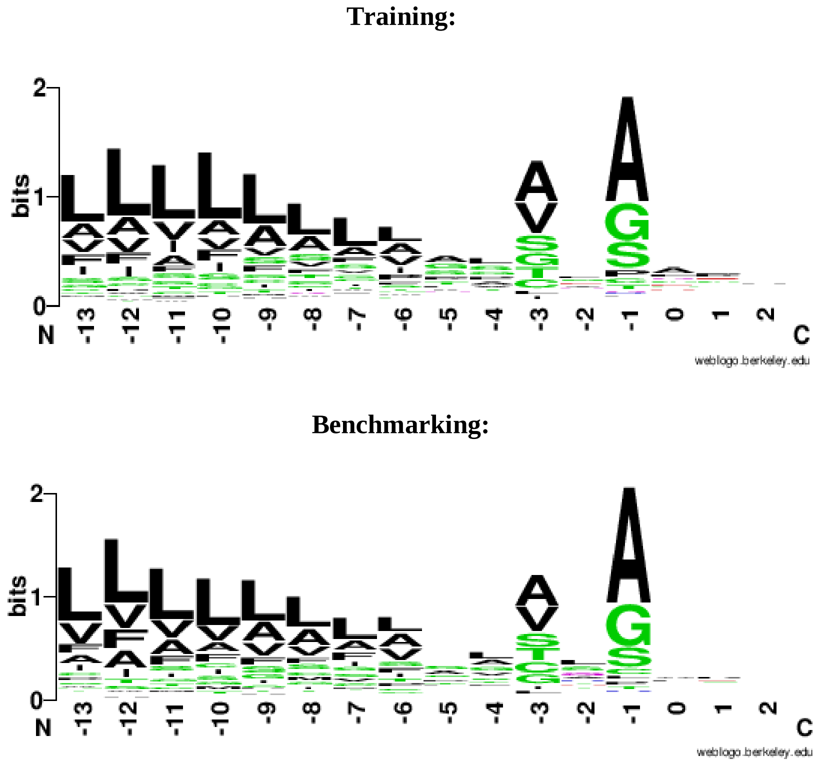


Figure 5: Amino Acid Composition Comparison

Sequence Logos of SP Cleavage Sites:

Sequence logos generated for SP cleavage sites, displaying the motifs from positions [-13,+2] in training and benchmarking datasets. The upper logo represents the training dataset, and the lower logo represents the benchmarking dataset. The logos illustrate the conserved patterns in signal peptide sequences, with a strong presence of Leucine (L) in the h-region (residues -13 to -6) and small, neutral residues like Alanine (A) and Glycine (G) in the c-region (residues -5 to -1). These motifs are crucial for accurate signal peptide detection and cleavage site prediction.


Training:

Benchmarking:

Figure 6: Sequence Logos of SP Cleavage Sites

Methods and Models

Von Heijne Method

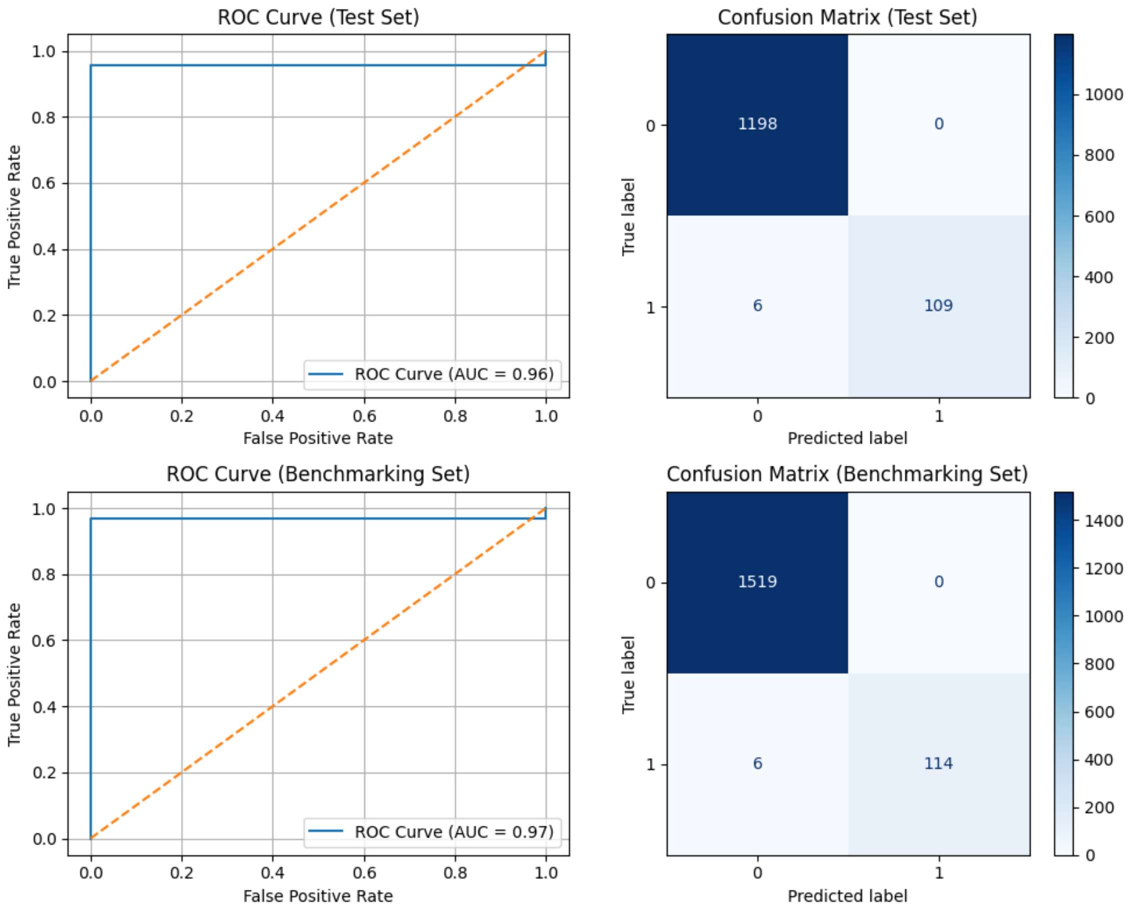
The von Heijne method, introduced by Gunnar von Heijne in 1986, is a pioneering rule-based approach for predicting signal peptides and their cleavage sites in protein sequences. This method utilizes a position-specific weight matrix (PSWM) to assign scores to amino acids at specific positions relative to the predicted cleavage site. The PSWM is constructed from empirical data on known signal peptides, capturing characteristic patterns such as the hydrophobic core and conserved motifs, notably the (-3, -1) rule (von Heijne, 1986).

Workflow

1. **Data Preparation:** Positive sequences containing known signal peptides and negative sequences lacking signal peptides were compiled from publicly available databases. The cleavage site positions and the surrounding 15-residue sequences were extracted. The combined dataset was then filtered, shuffled, and split into training, validation, and test sets.
2. **Creating the PSWM:** The PSWM was constructed by calculating the amino acid frequencies at each position within the signal peptide sequences from the positive dataset. These frequencies were then converted into log-odds scores using background amino acid frequencies derived from the SwissProt database.
3. **Model Training and Evaluation:** The PSWM was applied to the training, validation, and test sets, scoring each sequence according to its similarity to known signal peptides. The optimal threshold for classification was determined using the validation set. The model's performance was evaluated on the test set using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
4. **Blind Test Evaluation:** The model was applied to a separate benchmarking dataset to assess its robustness in real-world scenarios. Performance metrics confirmed the model's reliability.

5. **Visualizing Model Performance:** The performance of the von Heijne method was visualized through various metrics, including ROC curves and confusion matrices.

The von Heijne method demonstrated excellent performance on both the test and benchmarking datasets. The optimal threshold for classification was found to be 0.4385. On the test set, the model achieved an accuracy of 99.54%, precision of 100.00%, recall of 94.78%, and an F1 score of 97.32%. For the benchmarking dataset, the model achieved an accuracy of 99.63%, precision of 100.00%, recall of 95.00%, and an F1 score of 97.44%. The ROC curves further illustrate the model's outstanding discriminatory power, with AUCs of 0.96 for the test set and 0.97 for the benchmarking set.



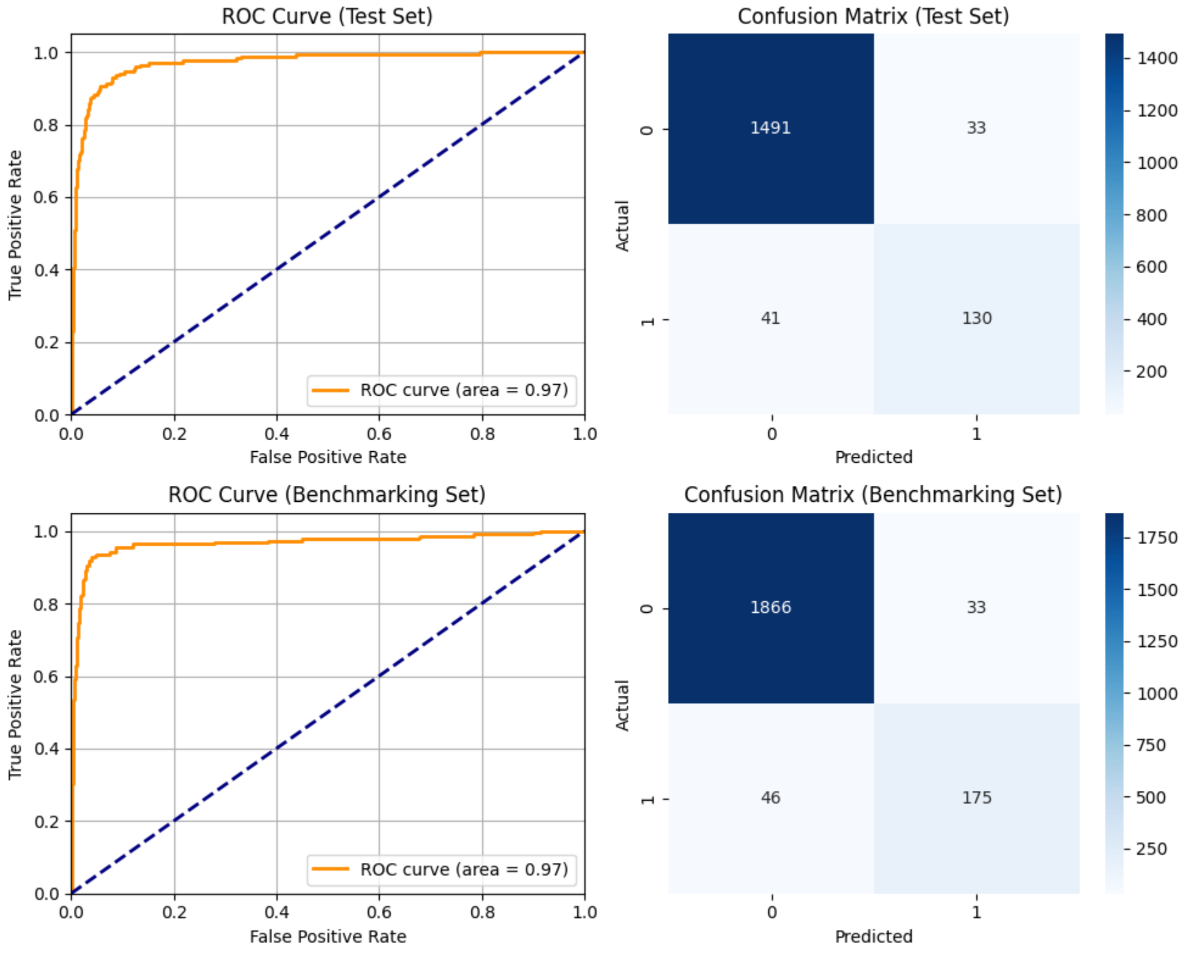
Support Vector Machine (SVM)

Support Vector Machines (SVM), introduced by Vladimir Vapnik in the 1990s, are a class of supervised learning models widely used for classification and regression tasks. SVMs operate by identifying the hyperplane that best separates the classes in a high-dimensional feature space, maximizing the margin between the support vectors, which are the closest points of each class to the hyperplane. This method is particularly effective for binary classification tasks, such as signal peptide detection (Vapnik, 1995).

Workflow

1. **Data Preparation:** The initial step involved loading the fasta and text files for both positive and negative training sets, as well as the benchmarking sets. The data was then divided into training, validation, and test subsets.
2. **Define Frequency Calculation Function:** A function was developed to calculate the amino acid frequencies for each sequence. These frequency vectors served as feature representations for the sequences.
3. **Generate Frequency Vectors:** Frequency vectors were generated for the training, validation, and testing subsets for various k-mer lengths.
4. **Train and Validate SVM Model:** The SVM model was trained using a combination of different subsets, validated with the fourth subset, and tested with the fifth. A grid search was conducted to optimize the hyperparameters, specifically C and gamma. The RBF kernel was selected for its effectiveness in handling nonlinear relationships.

The SVM model with the RBF kernel demonstrated strong performance across all datasets. The high accuracy (96%) and precision (84% on benchmarking data) indicate that the model effectively distinguishes between signal peptides and non-signal peptides with minimal false positives. The ROC curves for both the test and benchmarking sets display high AUC values (0.97), confirming the model's excellent discriminatory power.



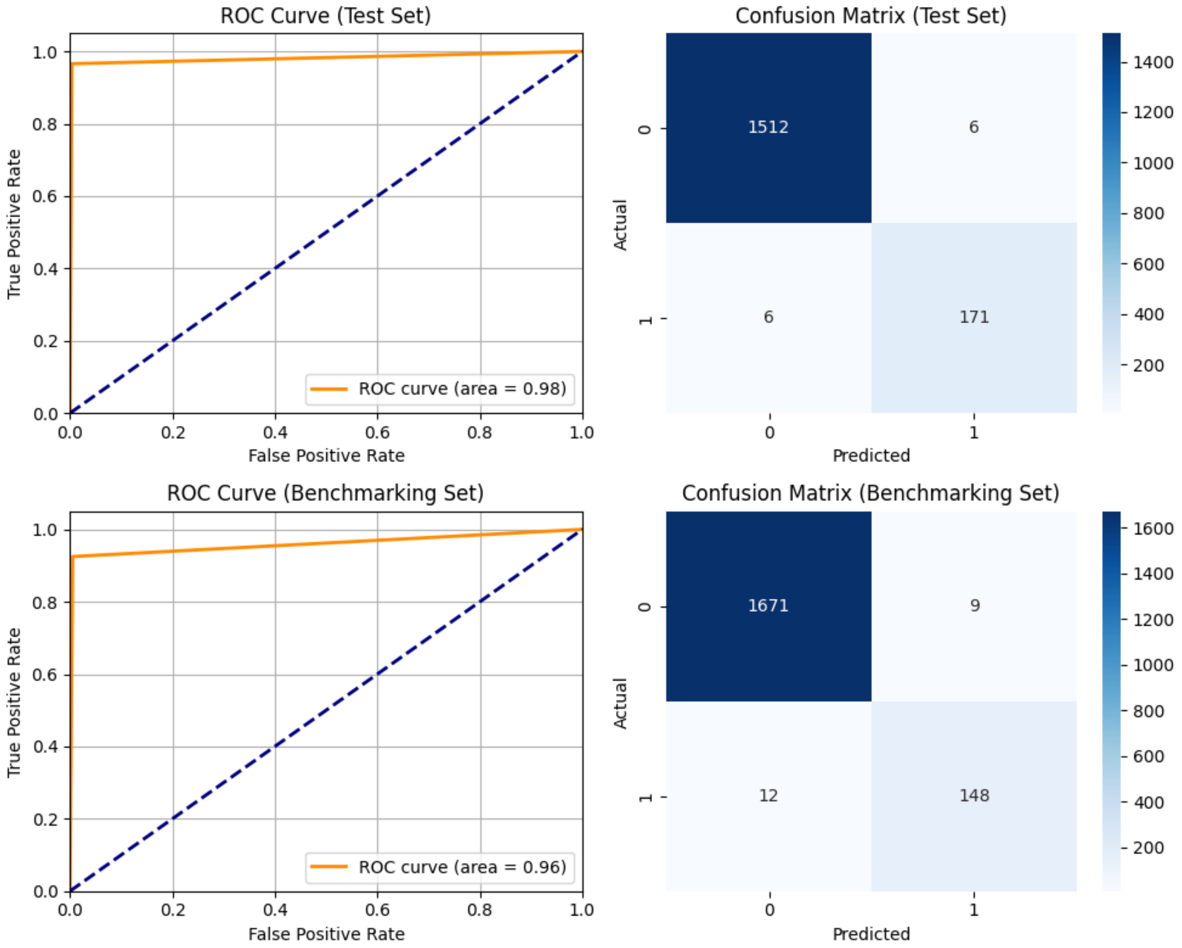
SignalP

SignalP is a widely used computational tool for predicting signal peptides and their cleavage sites in protein sequences. Initially developed by Nielsen et al. in 1997, SignalP has evolved through several versions, integrating advanced machine learning techniques to enhance its predictive accuracy. The latest versions of SignalP utilize deep learning models for enhanced prediction (Nielsen et al., 1997; Petersen et al., 2011).

Workflow

1. **Data Preparation:** Loaded the fasta and text files for both positive and negative training sets, as well as the benchmarking sets. The data was divided into training, validation, and test subsets.
2. **Neural Network Training:** The neural networks in SignalP are trained on large datasets of sequences with known signal peptides and cleavage sites. The training process involves adjusting the network's weights to minimize prediction error.

SignalP demonstrated outstanding performance on both the test and benchmarking datasets. The high accuracy (99.29% on the test set and 98.86% on the benchmarking set) indicates that the model reliably predicts signal peptides and their cleavage sites. The precision and recall values are consistently high, highlighting the model's ability to correctly identify true positives with minimal false positives and false negatives. The ROC curves further confirm the model's excellent discriminatory power.



DeepSig

DeepSig is a state-of-the-art deep learning-based approach specifically designed for signal peptide detection in proteins. Developed by Savojardo et al. in 2018, DeepSig employs deep convolutional neural networks (DCNNs) to significantly improve the accuracy of signal peptide identification (Savojardo et al., 2018).

Workflow

1. Deep Convolutional Neural Network (DCNN):

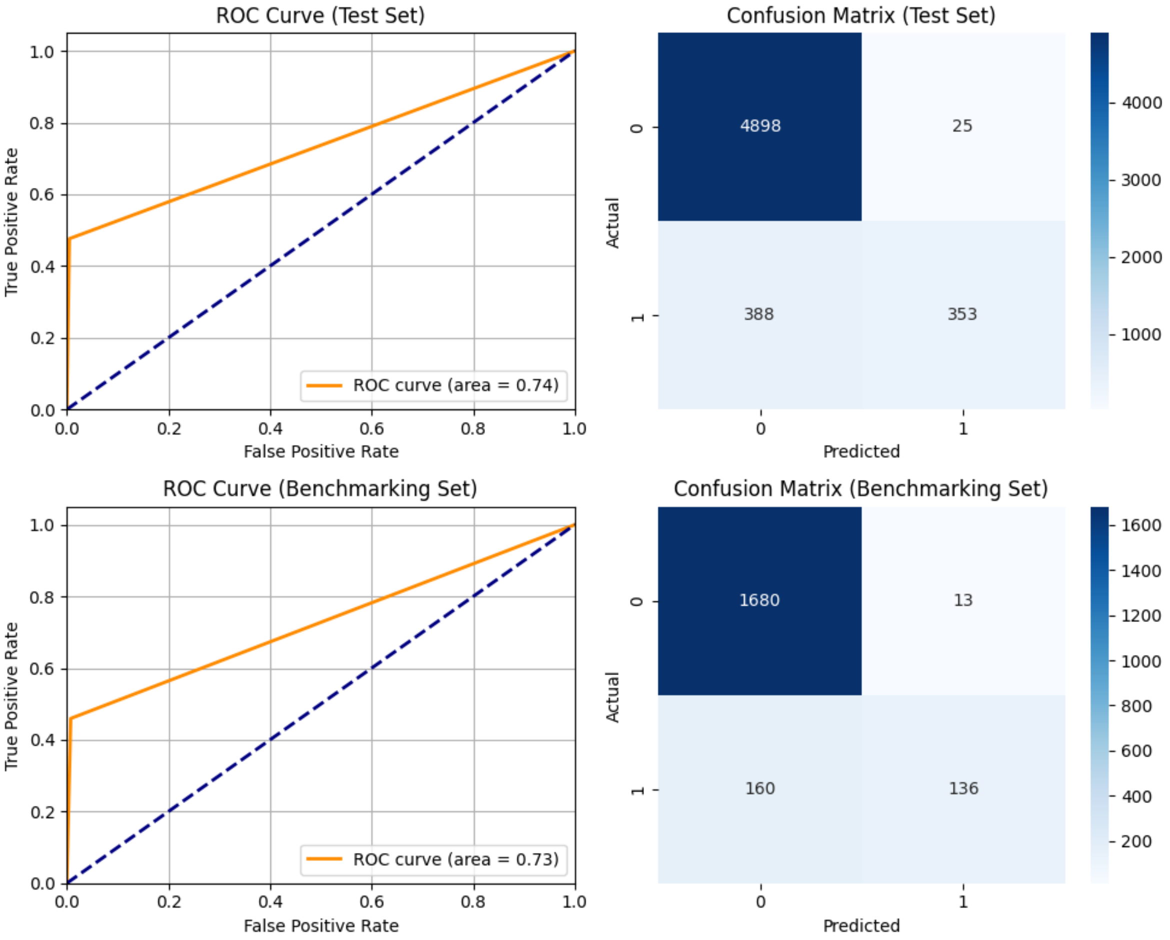
- **Convolutional Layers:** The DCNN in DeepSig is composed of three convolution-pooling stages. Convolutional layers apply filters to the input sequences to detect local patterns, such as motifs, which are crucial for identifying signal peptides.
- **Pooling Layers:** Pooling layers follow the convolutional layers to reduce the dimensionality of the feature maps. This step helps in summarizing the most important information and making the network more computationally efficient and robust to variations in the input data.
- **Fully Connected Layers:** After the convolution and pooling stages, the extracted features are passed through fully connected layers. These layers integrate the detected features and perform the final classification by assigning probabilities to each input sequence for being a signal peptide or not.

2. Training Process:

- **Dataset:** DeepSig is trained on a large dataset comprising both positive sequences (with known signal peptides) and negative sequences (without signal peptides). This comprehensive dataset ensures that the model learns a wide variety of patterns associated with signal peptides.

- **Backpropagation and Optimization:** The training process involves backpropagation, where the error is propagated back through the network to adjust the weights, and optimization techniques, such as gradient descent, to minimize the prediction error. This iterative process continues until the model's performance stabilizes and reaches an optimal point.

DeepSig demonstrated robust performance with an accuracy of 92.71% on the test set and 91.30% on the benchmarking set. The high precision values suggest that DeepSig effectively identifies true positives with minimal false positives. However, the recall values were lower, indicating that while the model is precise, it misses a significant number of true positive signal peptides. The ROC curves show good but not excellent discriminatory power, reflecting the model's strengths in precision but also highlighting areas for potential improvement in recall.



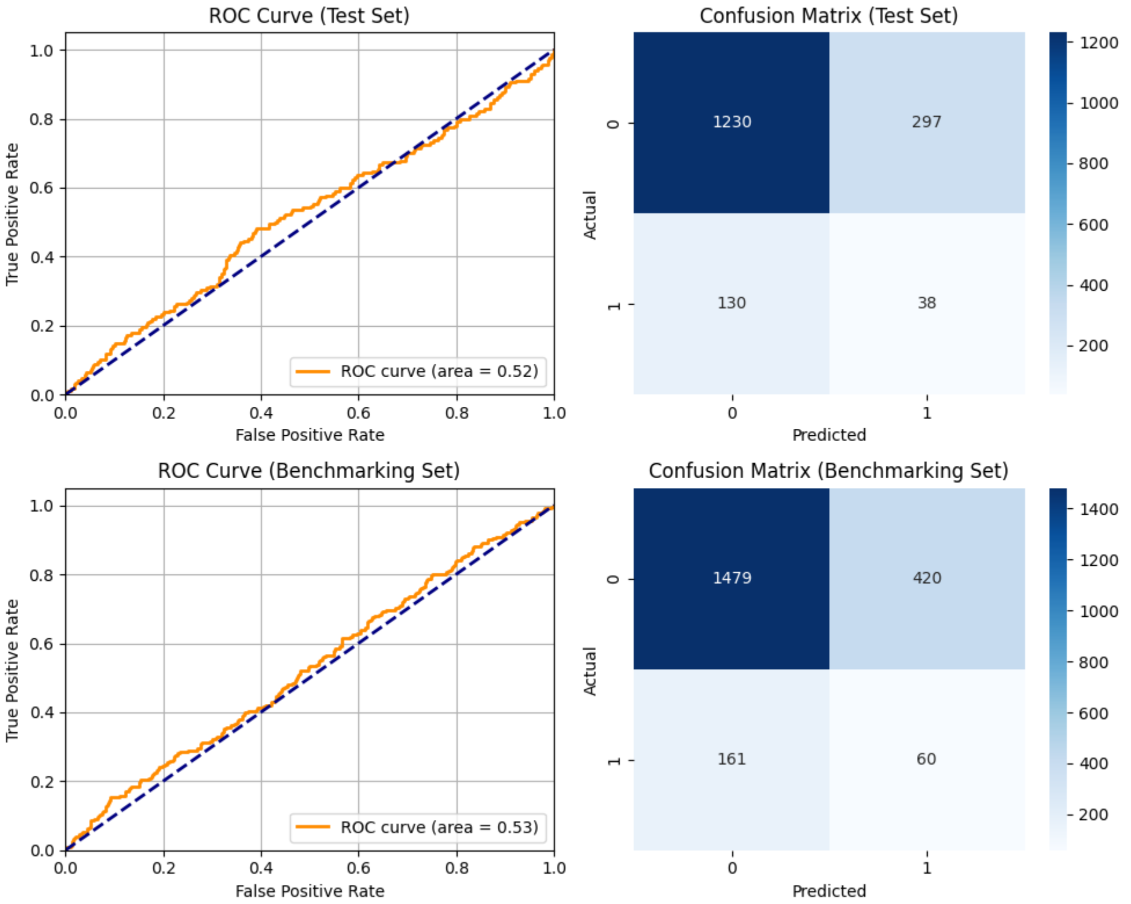
Fully Connected Neural Network (FCNN)

A Fully Connected Neural Network (FCNN), also known as a Multilayer Perceptron (MLP), is a type of artificial neural network where each neuron in one layer is connected to every neuron in the next layer (Goodfellow et al., 2016).

Workflow

1. **Data Preparation:** Sequences from positive and negative training sets were loaded from fasta files. The sequences were combined into a single DataFrame and shuffled.
2. **Sequence Encoding and Padding:** Sequences were encoded into numerical vectors and padded to a fixed length.
3. **Model Architecture:** The FCNN consisted of several fully connected layers with ReLU activation functions and dropout layers for regularization.
4. **Training and Optimization:** The model was trained using the Adam optimizer with binary cross-entropy loss function. Early stopping and a learning rate scheduler were implemented to optimize training.

The FCNN model showed an accuracy of 74.81% on the test set and 72.59% on the benchmarking set. Precision and recall metrics were notably low, indicating that the model struggled to correctly identify true positives. The ROC curves for both the test and benchmarking sets show AUC values close to 0.5, indicating that the model's predictions were not better than random chance.



Results Table

Model	Metric	Test Set	Benchmarking Set
Von Heijne	Accuracy	0.9954	0.9963
	Precision	1.0000	1.0000
	Recall	0.9478	0.9500
	F1 Score	0.9732	0.9744
SVM	Accuracy	0.9929	0.9886
	Precision	0.9661	0.9427
	Recall	0.9661	0.9250
	F1 Score	0.9661	0.9338
SignalP	Accuracy	0.9929	0.9886
	Precision	0.9661	0.9427
	Recall	0.9661	0.9250
	F1 Score	0.9661	0.9338
DeepSig	Accuracy	0.9271	0.9130
	Precision	0.9339	0.9128
	Recall	0.4764	0.4595
	F1 Score	0.6309	0.6112
FCNN	Accuracy	0.7481	0.7259
	Precision	0.1139	0.1254
	Recall	0.2262	0.2727
	F1 Score	0.1510	0.1721

Table 1: Performance Metrics of Various Signal Peptide Detection Models.

Discussion

The study highlights the importance of accurately detecting signal peptides in protein sequences using machine learning methods. Among the evaluated models, the von Heijne method demonstrated superior performance with near-perfect precision and high recall, underscoring its robustness in signal peptide detection. This rule-based approach benefits from established empirical knowledge, making it highly reliable.

The SVM and SignalP models both performed exceptionally well, leveraging the strengths of machine learning algorithms to identify complex patterns in the data. Their high precision and recall values highlight their effectiveness in accurately detecting signal peptides, indicating strong predictive power. High precision indicates a low rate of Type I errors (false positives), meaning that almost all identified signal peptides were indeed signal peptides. High recall suggests a low rate of Type II errors (false negatives), meaning that most true signal peptides were correctly identified. The F1 score, which balances precision and recall, was also high for these models, confirming their overall efficacy.

DeepSig, despite its advanced deep learning framework, underperformed compared to SVM and SignalP. This lower performance might be attributed to insufficient feature extraction or overfitting issues, which are common challenges in deep learning models without extensive hyperparameter tuning and large datasets. Overfitting occurs when a model learns the training data too well, including noise and outliers, which can degrade its performance on new, unseen data. The recall values for DeepSig were particularly lower, indicating that the model missed a significant number of true positive signal peptides, resulting in higher Type II errors. The high precision shows that when DeepSig predicted a signal peptide, it was usually correct, but the lower recall suggests it failed to detect many true signal peptides.

The FCNN model performed poorly, with metrics indicating it was no better than a random classifier. This suggests inadequate model complexity, poor feature representation, or insufficient training data. The FCNN's low precision and recall resulted in a low F1 score, showing it struggled to correctly identify true positives (high Type II errors) and often misclassified non-signal peptides as signal peptides (high Type I errors).

The size of the dataset, particularly for training deep learning models, may have been insufficient to capture the complex patterns required for accurate signal peptide detection. Larger datasets provide more examples and variations, which help models generalize better to new data. The FCNN model's architecture may not have been complex enough to learn the intricate features of the dataset, leading to poor performance. Limited tuning of hyperparameters might have constrained the performance of some models, particularly DeepSig and FCNN. Hyperparameter optimization is crucial for adjusting the model to better capture the underlying patterns in the data.

Future Directions

Enhanced Data Collection: Expanding the dataset with more diverse sequences could improve model training and performance, particularly for deep learning models. This would provide a broader range of examples for the model to learn from, enhancing its ability to generalize.

Advanced Model Architectures: Exploring more sophisticated architectures such as recurrent neural networks (RNNs) or transformer models could better capture the sequential nature of protein sequences. These models are designed to handle sequential data and could provide better performance in signal peptide detection.

Hyperparameter Optimization: Implementing comprehensive hyperparameter tuning techniques could optimize model performance, especially for complex models like DeepSig. Techniques such as grid search, random search, or Bayesian optimization can systematically explore the hyperparameter space to find the best settings.

Feature Engineering: Incorporating additional biologically relevant features might enhance model accuracy and robustness. Features such as evolutionary conservation, secondary structure, and physicochemical properties of amino acids could provide more information for the models to leverage.

Conclusion

This study underscores the critical role of accurately detecting signal peptides in protein sequences using machine learning methods. The von Heijne method, along with SVM and SignalP, demonstrated high reliability and precision, making them valuable tools in bioinformatics. DeepSig and FCNN, while innovative, require further refinement to achieve comparable performance. Future research should focus on improving data diversity, model architectures, and hyperparameter optimization to advance the field of signal peptide detection. The continued development and application of machine learning in bioinformatics hold significant potential for enhancing our understanding of protein sorting and compartmentalization. Balancing precision and recall through methods like the F1 score remains crucial to ensure robust and reliable predictions in bioinformatics applications.

Supplementary Material

All the codes and files used in this project are provided in the following GitHub link:

https://github.com/nafisaaa1/Laboratory_Bioinformatics2

References

- *von Heijne, G. (1983). "Patterns of amino acids near signal-sequence cleavage sites." European Journal of Biochemistry, 133(1), 17-21.*
- *von Heijne, G. (1990). "The signal peptide." Journal of Membrane Biology, 115(3), 195-201.*
- *Nugent, T., & Jones, D. T. (2009). "Transmembrane protein topology prediction using support vector machines." BMC Bioinformatics, 10(1), 159.*
- *Petersen, T. N., Brunak, S., von Heijne, G., & Nielsen, H. (2011). "SignalP 4.0: discriminating signal peptides from transmembrane regions." Nature Methods, 8(10), 785-786.*
- *Savojardo, C., Martelli, P. L., Fariselli, P., Casadio, R., & Bo, L. (2018). "DeepSig: deep learning improves signal peptide detection in proteins." Bioinformatics, 34(10), 1690-1696.*
- *Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." MIT Press.*
- *Martoglio, B., & Dobberstein, B. (1998). "Signal sequences: more than just greasy peptides." Trends in Cell Biology, 8(10), 410-415.*
- *Choo, K. H. A., Tan, T. W., & Ranganathan, S. (2009). "A comprehensive assessment of N-terminal signal peptides prediction methods." BMC Bioinformatics, 10(Suppl 15), S1.*
- *von Heijne, G. (1986). "A new method for predicting signal sequence cleavage sites." Nucleic Acids Research, 14(11), 4683-4690.*
- *Nielsen, H., Engelbrecht, J., Brunak, S., & von Heijne, G. (1997). "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites." Protein Engineering, 10(1), 1-6.*
- *Nakai, K., & Kanehisa, M. (1992). "A knowledge base for predicting protein localization sites in eukaryotic cells." Genomics, 14(4), 897-911.*
- *SwissProt Database: SwissProt - A curated protein sequence database providing a high level of annotation, including the function of proteins, domains structure, post-translational modifications, and variants.*
- *Vapnik, V. (1995). "The Nature of Statistical Learning Theory." Springer.*
- *Nielsen, H., Engelbrecht, J., Brunak, S., & von Heijne, G. (1997). "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites." Protein Engineering, 10(1), 1-6.*
- *Petersen, T. N., Brunak, S., von Heijne, G., & Nielsen, H. (2011). "SignalP 4.0: discriminating signal peptides from transmembrane regions." Nature Methods, 8(10), 785-786.*
- *Almagro Armenteros, J. J., Tsirigos, K. D., Sønderby, C. K., Petersen, T. N., Winther, O., Brunak, S., ... & Nielsen, H. (2019). "SignalP 5.0 improves signal peptide predictions using deep neural networks." Nature Biotechnology, 37(4), 420-423.*
- *Savojardo, C., Martelli, P. L., Fariselli, P., & Casadio, R. (2018). "DeepSig: deep learning improves signal peptide detection in proteins." Bioinformatics, 34(10), 1690-1696.*