picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Detecting Signal Peptide in Proteins: a comparisoin between Support Vector Machine and vonHeijne Method

Maryam Mohammadi

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Detecting signal peptides in proteins is an essential task in bioinformatics and computational biology. Identifying signal peptides is very important because they play a crucial role in protein targeting and secretion. Accurate detection of signal peptides can provide valuable insights into protein function and localization and ofcourse further improvements and advancements in signal peptide detection methods will undoubtedly enhance our understanding of protein biology and facilitate various biological studies. accuracy.

Results: To predict signal peptide regions in the proteins we trained two algorithm (VonHeijn and SVM). the performance evaluation of the SVM and von Heijne methods demonstrate that both method show promising performance in signal peptide detection. SVM achieves a MCC of 0.7466 and F1 score of 0.7438; these resulted metric for von Heijne method was lower (MCC of 0.5905 and F1 score of 0.5909). Based on the obrained results, the SVM method appears to outperform the VH algorithm in detecting signal peptides means It demonstrates higher values across multiple evaluation metrics, including MCC, Q2, precision, recall, and F1 Score. It is noteworthy to mention that the results contribute to the understanding of the strengths and limitations of these methods and can guide researchers in selecting the most suitable approach for their specific applications.

Availability: All of used material such as training dataset and test dataset with the implementationas are available in the email attach.

Contact: Maryam.Mohammadi3@studio.unibo.it

Supplementary information: Supplementary data are available at *supplementary file* .

1 Introduction

Signal peptides are short amino acid sequences found at the N-terminus of many secretory and membrane proteins. They play a crucial role in directing the newly synthesized proteins to their correct subcellular locations, such as the endoplasmic reticulum (ER), Golgi apparatus, or the cell membrane. The presence of a signal peptide is essential for the proper translocation and localization of proteins within the cell [1]. Signal peptides are typically located at the N-terminus of proteins and this feature allows for their recognition and subsequent targeting by the signal recognition particle (SRP) and translocon machinery. Signal peptides contain a hydrophobic core composed of amino acids, often referred to as the signal peptide "cleavage site." This hydrophobic region interacts with the translocation machinery during protein translocation.

Detection of Signal Peptide has some applications that we mention them here:

Protein Localization: Signal peptides determine the intracellular destination of proteins, enabling them to be correctly transported to their target organelles.

Secretory Pathway Analysis: Studying signal peptides provides insights into the secretory pathway.

Drug Target Identification: Many pharmaceutical drugs target membrane proteins. Detecting signal peptides aids in identifying potential

© The Author 2015. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

drug targets and understanding their transport and insertion into cellular membranes.

Biotechnology and Protein Engineering: Signal peptides are utilized in recombinant protein production for efficient secretion and proper folding of heterologous proteins.

Characteristics of Signal Peptides: Signal peptides possess specific characteristics that make them distinguishable from other regions of proteins like: Conserved Motifs: Signal peptides exhibit conserved sequence motifs and structural features, such as the presence of a positively charged N-region, a hydrophobic H-region, and a polar C-region (These motifs aid in their recognition and binding to specific cellular components)

Challenges in Signal Peptide Prediction: While the detection of signal peptides has proven to be a valuable tool, it still presents some challenges to predict it:

Sequence Variability: Signal peptides can exhibit considerable sequence variability, making it difficult to identify common patterns across different proteins. This variability requires the development of robust computational methods that can handle diverse signal peptide sequences.

Signal Peptide Overlap: Some proteins may have multiple signal peptides or signal peptide-like sequences. Distinguishing between functional signal peptides and non-functional or alternative signal sequences is a challenge that requires careful analysis.

Overall, the detection and prediction of signal peptides in proteins are crucial for understanding protein localization, secretory pathways, and protein engineering. Despite the challenges, advancements in computational algorithms and experimental techniques have improved the accuracy of signal peptide prediction.

2 Material and methods

2.1 Datasets

2.1.1 Training Dataset

The training dataset used in this project consists of 1,723 proteins which extracted from the UniProtKB database with 258 proteins containing signal peptides. This dataset allows for the training models by using the von Heijne algorithm and SVM method. In this section, we will discuss the characteristics of the training dataset, including signal peptide length distribution, amino acid composition, and taxonomic classification.

Signal Peptide Length Distribution: The signal peptide length distribution provides insights into the range and variability of signal peptide lengths in the training dataset. Upon analysis (Figure-1), the median signal peptide length was found to be 21 residues, with a variance of 22.45.

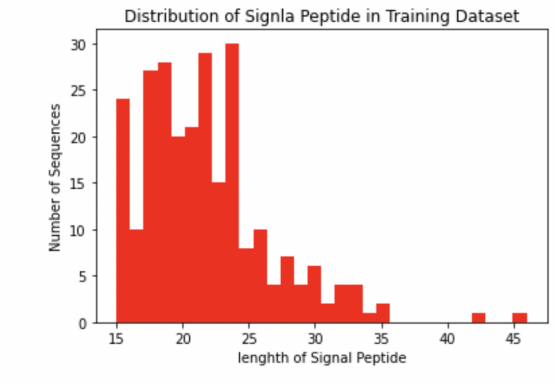

Fig. 1: distribution of signal peptide length in training dataset

About .75 of signal peptides has the lenght less than 24 residues, while the maximum and minimum signal peptide lengths were 46 and 15 residues, respectively. This information indicates the diversity and distribution of signal peptide lengths in the dataset.

Amino Acid Composition in Signal Peptide Region: Analyzing the amino acid composition in the signal peptide region of proteins provides valuable information about the prevalence of specific amino acids in the singla peptide region. Comparing these percentages with the overall composition of swissprot database provides insights into the bias or difference in amino acid occurrence within the signal peptide region.The shown percentages highlight the relative abundance of each amino acid in the signal peptide region. For example, Leucine (L) is the most abundant amino acid, with an occurrence percentage of 23.19% while in the overall SwissProt dataset, the highest occurrence is observed for Leucine (L) at 9.65%. This difference suggests a potential bias or specificity of certain amino acids in the signal peptide region of the training dataset. Figure-2; and aslo the aminoacids like Lucine and Alanine (some non-polar aminoacids) are overrepresented in the training dataset.

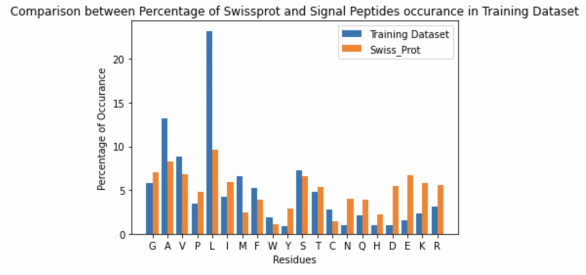

Fig. 2: Comparison between the composition of signal peptide of trainig dataset and the whole swissprot database

Taxonomic Classification in the Training Dataset: The taxonomic classification of proteins in the training dataset helps understand the distribution of signal peptides across different kingdoms. As you can see in the Figure-3 the classification revealed the presence of signal peptide proteins from four kingdoms: Fungi (14 proteins), Metazoa (215 proteins), Other (1 protein), and Plants (28 proteins). This information highlights the taxonomic diversity within the dataset and allows for further analysis of signal peptides in different biological contexts.

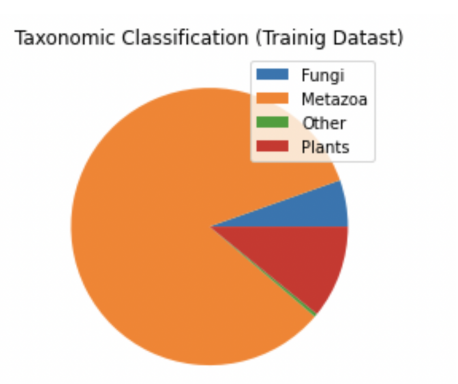

Fig. 3: distriubtion of signal peptide across different kingdom

2.1.2 Blind Dataset

The benchmark dataset used in this study consists of 7,456 proteins which extracted from the UniProtKB database and include 209 signal peptide proteins. This dataset serves as a validation set for evaluating the performance of the trained prediction models of von Heijne algorithm and SVM method. In this section, we will discuss the characteristics of the benchmark dataset, including signal peptide length distribution, amino acid composition, and taxonomic classification.

Signal Peptide Length Distribution: Analyzing the signal peptide length distribution in the benchmark dataset provides insights into the length variability and range of signal peptides present. As you can see in the Figure-4 the median signal peptide length was found to be 23 residues, with a variance of 37.38; and 75% of signal peptide length are less than 27 residues, while the maximum and minimum signal peptide lengths were 46 and 13 residues, respectively. These statistics indicate the diversity of signal peptide lengths in the benchmark dataset.

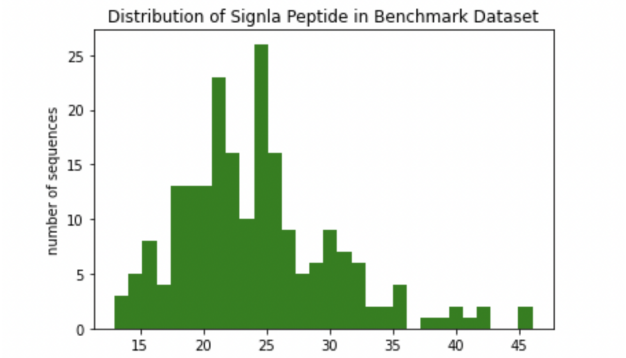

Fig. 4: distriubtion of signal peptide length in Benchmark dataset

Amino Acid Composition in Signal Peptide Region: Comparing the amino acid compositions between the benchmark dataset and the overall SwissProt dataset shows variations in the prevalence of amino acids in the signal peptide region. For instance,in the Figure-5 the benchmark dataset exhibits a higher occurrence percentage of Alanine (A) at 13.88% compared to the overall SwissProt dataset, where it is 8.25%. These differences may indicate the specific requirements or functional relevance of certain amino acids in the signal peptide region within the benchmark dataset.

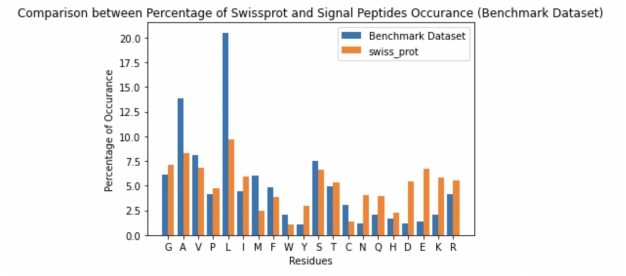

Fig. 5: comparison between composition of signal peptide region in Benchmark dataset and the whole swissprot database

Taxonomic Classification in the Benchmark Dataset: The taxonomic classification of proteins in the benchmark dataset provides insights into the distribution of signal peptides across different kingdoms. The classification revealed the presence of proteins (Figure-6) from four kingdoms: Fungi (20 proteins), Metazoa (169 proteins), Other (3 proteins), and Plants (17 proteins). This taxonomic diversity within the benchmark dataset allows for the evaluation of signal peptide prediction performance across different biological contexts and evolutionary lineages. This information is essential for developing accurate prediction models

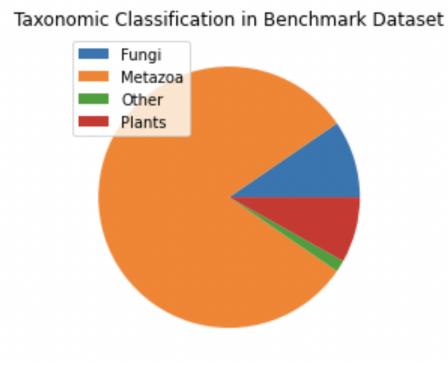

Fig. 6: comparison of signal peptide distribution between diffrent kingdom

and understanding the functional relevance of signal peptides in protein targeting and secretion. Moreover, understanding the characteristics, biases, and differences in signal peptide datasets is crucial for developing reliable prediction algorithms. By considering the specific amino acid compositions and taxonomic distributions, researchers can refine their computational models and interpret the results in a more accurate and context-specific manner. The findings from this analysis contribute to the knowledge and understanding of signal peptides and their functional relevance in protein targeting and secretion.

2.2 vonHeijne Method

The von Heijne algorithm, also known as the signal peptide prediction algorithm, is a widely used computational method for identifying signal peptides in proteins. This method developed by Gunnar von Heijne in 1983, the algorithm utilizes sequence-based features to predict the presence and location of signal peptides, which play a crucial role in protein targeting and secretion processes.

The algorithm operates on the assumption that signal peptides possess specific sequence patterns that distinguish them from the rest of the protein sequence. These patterns are associated with key structural elements and functional characteristics of signal peptides, such as the presence of a hydrophobic core, a signal peptidase cleavage site, and a positively charged region.(Figure-7)

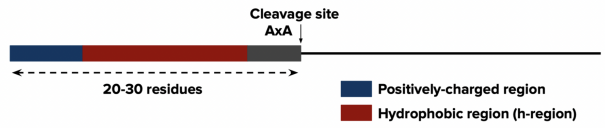

Fig. 7: structure of signal peptide segment in the SP proteins: consisting of hydrophobic core, signal peptide cleavage site and positive charge region

To detect signal peptides, the von Heijne algorithm employs a sliding window approach, where a window of fixed length is moved along the protein sequence, examining the properties of the window at each position. The algorithm calculates a prediction score based on the observed features within the window and compares it to a predefined threshold. If the score exceeds the threshold, the window is classified as a potential signal peptide [5].

One of the critical features used by the algorithm is the hydrophobicity profile. Signal peptides typically contain a hydrophobic core, which serves as a signal for protein translocation. The algorithm calculates the hydrophobicity score by assigning a hydrophobicity value to each amino acid residue and summing these values within the window. Regions with high hydrophobicity scores are indicative of potential signal peptides. Another important feature considered by the von Heijne algorithm is the presence of a signal peptidase cleavage site. Signal peptides are typically cleaved by signal peptidases, resulting in the removal of the signal peptide and the release of the mature protein. The algorithm examines the amino acid residues surrounding the predicted cleavage site and evaluates their sequence patterns to determine the likelihood of cleavage. The algorithm evaluates the distribution and frequency of these residues to enhance its prediction accuracy.

The von Heijne algorithm has been widely utilized in various studies and applications. It has been applied to large-scale proteomic analyses to identify proteins with signal peptides and classify them into different subcellular compartments or secretion pathways also.

2.2.1 vonHeijne implementation

Data Preparation:

Collect a dataset of proteins that are known to have signal peptides (SPs) and then extracting the region around the cleavage sites, covering 15 residues from position -13 to +2 relative to the cleavage site[6] .Figure-8

Fig. 8: Modeling the region around cleavage site of signal peptide proteins

Divide the dataset into training and testing sets:

To train the VonHeijn algorightm, first the Position-Specific Probability Matrix (PSPM) should be calcualted from the training set(Figure-9). So , we Initialize the PSPM matrix with pseudocounts (add 1 to each position to avoid zero probabilities) and Iterate over the training data and update the PSPM by counting the occurrences of each residue at each position.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0.068345	0.118705	0.129496	0.093525	0.18705	0.125899	0.089928	0.147482	0.161871	0.089928	0.244604	0.071942	0.456835	0.129496	0.05036
R	0.007194	0.007194	0.003597	0.003597	0.003597	0.003597	0.021583	0.007194	0.028777	0.035971	0.007194	0.068345	0.010791	0.043165	0.061151
N	0.003597	0.007194	0.003597	0.014388	0.010791	0.017986	0.007194	0.007194	0.021583	0.010791	0.010791	0.014388	0.003597	0.035971	0.035971
D	0.007194	0.003597	0.003597	0.003597	0.010791	0.003597	0.003597	0.017986	0.057554	0.010791	0.007194	0.05036	0.014388	0.086331	0.064748
C	0.028777	0.02518	0.02518	0.021583	0.046763	0.05036	0.068345	0.039568	0.010791	0.028777	0.046763	0.017986	0.039568	0.017986	0.032374
Q	0.010791	0.007194	0.007194	0.010791	0.007194	0.014388	0.017986	0.014388	0.053957	0.035971	0.007194	0.093525	0.014388	0.05036	0.039568
E	0.003597	0.003597	0.003597	0.007194	0.007194	0.010791	0.028777	0.003597	0.028777	0.021583	0.007194	0.115108	0.007194	0.05036	0.068345
G	0.028777	0.021583	0.02518	0.032374	0.028777	0.028777	0.046763	0.043165	0.111511	0.154676	0.093525	0.039568	0.179856	0.079137	0.053957
H	0.003597	0.014388	0.007194	0.003597	0.003597	0.003597	0.014388	0.007194	0.028777	0.017986	0.007194	0.043165	0.003597	0.021583	0.035971
I	0.089928	0.064748	0.086331	0.035971	0.064748	0.028777	0.039568	0.064748	0.02518	0.021583	0.035971	0.028777	0.010791	0.02518	0.039568
L	0.406475	0.392086	0.348921	0.410072	0.294964	0.359712	0.338129	0.226619	0.064748	0.158273	0.02518	0.118705	0.017986	0.07554	0.046763
K	0.003597	0.003597	0.003597	0.003597	0.003597	0.003597	0.003597	0.003597	0.021583	0.007194	0.007194	0.035971	0.010791	0.068345	0.046763
M	0.021583	0.017986	0.032374	0.017986	0.017986	0.028777	0.035971	0.028777	0.010791	0.021583	0.003597	0.014388	0.003597	0.010791	0.003597
F	0.082734	0.071942	0.057554	0.097122	0.089928	0.07554	0.053957	0.05036	0.02518	0.02518	0.003597	0.039568	0.007194	0.053957	0.039568
P	0.007194	0.021583	0.007194	0.014388	0.010791	0.017986	0.02518	0.089928	0.057554	0.104317	0.007194	0.010791	0.035971	0.010791	0.154676
S	0.046763	0.05036	0.053957	0.028777	0.068345	0.07554	0.071942	0.057554	0.111511	0.079137	0.125899	0.089928	0.125899	0.07554	0.061151
T	0.017986	0.02518	0.028777	0.046763	0.021583	0.043165	0.032374	0.039568	0.089928	0.068345	0.122302	0.028777	0.039568	0.05036	0.064748
W	0.017986	0.02518	0.028777	0.010791	0.010791	0.02518	0.028777	0.017986	0.014388	0.010791	0.003597	0.039568	0.007194	0.007194	0.003597
Y	0.017986	0.003597	0.003597	0.007194	0.007194	0.007194	0.003597	0.010791	0.010791	0.014388	0.003597	0.043165	0.003597	0.032374	0.021583
V	0.125899	0.115108	0.140288	0.136691	0.104317	0.07554	0.068345	0.122302	0.064748	0.082734	0.230216	0.035971	0.007194	0.07554	0.07554

Fig. 9: PSPM scores of training dataset (after finding the optimal threshold by the help of 5-k fold cross-validtion)

Normalize the PSPM by dividing each cell by the total number of sequences and then calculating the Position-Specific Weight Matrix (PSWM) from the PSPM devided by obtained background model(SwissProt AA composition) and then compute the log-odds ratio between the PSPM and the background model to get the PSWM. Figure-10

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	-0.22716	0.569309	0.69484	0.225355	1.225355	0.654198	0.168771	0.882467	1.016768	0.168771	1.612378	-0.15316	2.5136	0.69484	-0.66773
R	-3.06005	-3.06005	-4.06005	-4.06005	-4.06005	-4.06005	-1.47509	-3.06005	-1.06005	-0.73812	-3.06005	0.18788	-2.47509	-0.47509	0.027415
N	-3.47509	-2.47509	-3.47509	-1.47509	-1.89012	-1.15316	-2.47509	-2.47509	-0.89012	-1.89012	-1.89012	-1.47509	-3.47509	-0.15316	-0.15316
D	-3.06005	-4.06005	-4.06005	-4.06005	-2.47509	-4.06005	-4.06005	-1.73812	-0.06005	-2.47509	-3.06005	-0.25269	-2.06005	0.524915	0.109878
C	1.524915	1.33227	1.33227	1.109878	2.225355	2.33227	2.772843	1.984347	0.109878	1.524915	2.225355	0.846843	1.984347	0.846843	1.69484
Q	-1.89012	-2.47509	-2.47509	-1.89012	-2.47509	-1.47509	-1.15316	-1.47509	0.431806	-0.15316	-2.47509	1.225355	-1.47509	0.33227	-0.01565
E	-4.28244	-4.28244	-4.28244	-3.28244	-3.28244	-2.69748	-1.28244	-4.28244	-1.28244	-1.69748	-3.28244	0.71756	-3.28244	-0.47509	-0.03451
G	-1.28244	-1.69748	-1.47509	-1.11252	-1.28244	-1.28244	-0.582	-0.69748	0.671757	1.143825	0.418	-0.82301	1.361416	0.176992	-0.37555
H	-2.47509	-0.47509	-1.47509	-2.47509	-2.47509	-2.47509	-0.47509	-1.47509	0.524915	-0.15316	-1.47509	1.109878	-2.47509	0.109878	0.846843
I	0.583809	0.109878	0.524915	-0.73812	0.109878	-1.06005	-0.60062	0.109878	-1.25269	-1.47509	-0.73812	-1.06005	-2.47509	-1.25269	-0.60062
L	2.023166	1.971171	1.8029	2.035877	1.560539	1.846843	1.755576	1.180267	-0.62709	0.662419	-1.98966	0.247381	-2.47509	-0.4047	-1.09657
K	-4.06005	-4.06005	-4.06005	-4.06005	-4.06005	-4.06005	-4.06005	-4.06005	-1.47509	-3.06005	-3.06005	-0.73812	-2.47509	0.18788	-0.35961
M	0.109878	-0.15316	0.69484	-0.15316	-0.15316	0.524915	0.846843	0.524915	-0.89012	0.109878	-2.47509	-0.47509	-2.47509	-0.89012	-2.47509
F	1.048477	0.846843	0.524915	1.279803	1.168771	0.917233	0.431806	0.33227	-0.66773	-0.66773	-3.47509	-0.01565	-2.47509	0.431806	-0.01565
P	-2.79701	-1.21205	-2.79701	-1.79701	-2.21205	-1.47509	-0.98966	0.846843	0.202987	1.060968	-2.79701	-2.21205	-0.47509	-2.21205	1.629252
S	-0.582	-0.47509	-0.37555	-1.28244	-0.03451	0.109878	0.039488	-0.28244	0.671757	0.176992	0.846843	0.361416	0.846843	0.109878	-0.19498
T	-1.47509	-0.98966	-0.79701	-0.09657	-1.21205	-0.21205	-0.62709	-0.33758	0.846843	0.450915	1.29045	-0.79701	-0.33758	0.010342	0.372912
W	0.846843	1.33227	1.524915	0.109878	0.109878	1.33227	1.524915	0.846843	0.524915	0.109878	-1.47509	1.984347	-0.47509	-0.47509	-1.47509
Y	-0.73812	-3.06005	-3.06005	-2.06005	-2.06005	-2.06005	-3.06005	-1.47509	-1.47509	-1.06005	-3.06005	0.524915	-3.06005	0.109878	-0.47509
V	0.846843	0.71756	1.002962	0.965488	0.575541	0.109878	-0.03451	0.805023	-0.11252	0.241122	1.71756	-0.96051	-3.28244	0.109878	0.109878

Fig. 10: PSWM scores of training dataset (after finding the optimal threshold by the help of 5-k fold cross-validtion

VonHeijn method Prediction:

For each sequence in the testing set we should slide the PSWM over the sequence and calculating a score for each position and then we determine the maximum score among all positions. In the next step, we should assign a class label based on a threshold, means If the maximum score exceeds the threshold, predict the sequence as having an SP (positive class), otherwise, predict the sequence as not having an SP (negative class) and then the result of prediction will be stored for evaluation.

VonHeijn Threshold Selection:

There are two ways for selecting the threshold, the first one is using fixed threshold, means choosing a predefined threshold value (e.g., 0 or 5) based on domain knowledge or prior experiments; the second way is doing Cross-Validation, means optimizing the threshold in each training run using performance metrics such as precision-recall values, that for finding the optimal threshold we used the 5-fold cross-validation. Divide your dataset into five approximately equal-sized subsets (folds) labeled as Fold 1, Fold 2, ..., Fold 5. For each fold we designate the current fold as the validation set then combine the remaining four folds to create the training set (using the training procedure described earlier). For each threshold value we apply the trained model to the validation set and obtain predictions for each sequence then we compare the predictions with the true labels in the validation set then we calculate the desired performance metric (e.g., precision, recall, F1 score) for each threshold value. in the next step we should choose the threshold for each fold means we select the threshold value that maximizes the chosen performance metric on the validation set. This process is repeated for all five folds, resulting in five different threshold values. then we apply the average of these five folds threshold as the avg_threshold on BM dataset to evaluate the performance of trained model.

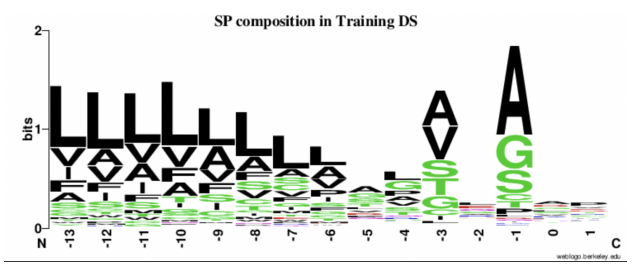

Fig. 11: motif for logo Training dataset

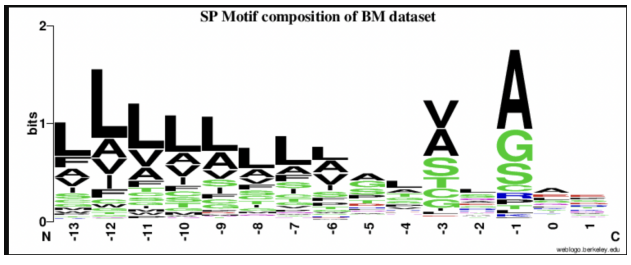

Fig. 12: motif for log Benchmark dataset

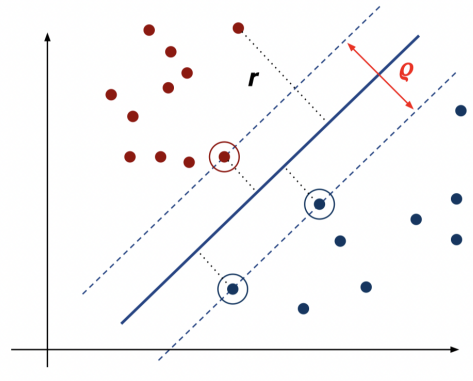

Fig. 14: Binary classification in SVM by support vectors of each classes

2.3 Support Vector Machine (SVM) Method

To overcome some limitations of VH method, we can use machine learning techniques, such as support vector machines (SVMs) models as an additional techniques. SVMs can capture more relationships between sequence patterns and improve the accuracy of signal peptide predictions. Ofcourse, the Von Heijne algorithm is a fundamental tool for the prediction of signal peptides in protein sequences. By considering sequence-based features, such as hydrophobicity, signal peptidase cleavage sites, and positively charged residues, the algorithm can identify potential signal peptides and aid in understanding protein targeting and secretion processes. Ongoing research continues to refine and enhance signal peptide prediction models, incorporating advanced computational methods and machine learning techniques to improve accuracy and reliability.

2.3.1 SVM Implementation

Support Vector Machines (SVMs) are supervised machine learning models used for both classification and regression problems. It introduced at the beginning of the 1990s by Vapnik as a supervised algorithm for classification [3].

For binary classification problems the aim is to find the best separating hyperplane, the one able to maximize the distance, called margin, between the nearest points of the two classes Figure-13

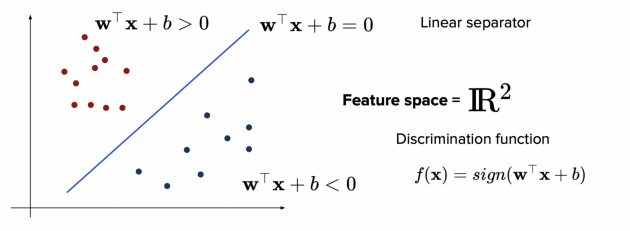

Fig. 13: Binary classification by separating hyperplane with the maxiumum margin

The points lying on the margins are called support vectors, these points are the only points that matter in our training dataset. Figure-14

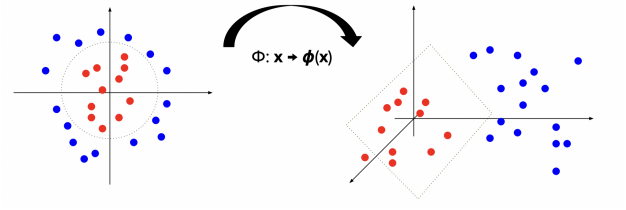

Fig. 15: Non-linear Classification and Kernel Trick: Mapping data to higher dimensional feature space and finding its hyperplane

Support Vector Machines (SVMs) has the capability to effectively address non-linear classification tasks by utilizing the kernel trick, which allows them to implicitly map data from the input space to a higher-dimensional feature space Figure-15. This property enables SVMs to efficiently handle large input spaces and effectively separate data into distinct classes [2]. Moreover, compared to other machine learning techniques, SVMs demonstrate a reduced susceptibility to overfitting, making them a favorable choice for modeling and classification tasks.

We first define SVM for linearly separable problem. Given a training dataset of n elements each having a value x e R and belonging to a class y -1, +1, the hyperplane is defined as equ.1:

$$\langle \mathbf{w}, \mathbf{x}_i \rangle + b = 0 \tag{1}$$

With w being the vector perpendicular to each point of the separating hyperplane. The distance of each point x ifrom the hyperplane is equ.2:

$$r = \left| \frac{\langle \mathbf{w}, \mathbf{x}_i \rangle + b}{\|\mathbf{w}\|} \right| \tag{2}$$

Defining the margin as we can state for all the points belonging to the either one of the y iclasses that equ.3:

$$y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \geq \rho/2 \tag{3}$$

Moving to only the support vectors x sthe inequality becomes an equality.

The margin can then be written as:
Eventually the optimization problem is the following equ.4:

$$minimize\frac{1}{2}\|\mathbf{w}^2\|with\left(\langle\mathbf{w},\mathbf{x}_i\rangle+b\right)\geq\ 1 \tag{4}$$

To solve it we exploit a Lagrange multiplier associated with every inequality constraint in the original problem [1]. We end up in solving a Lagrangian dual problem: equ.5:

$$\text{maximize}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left\langle\boldsymbol{x_i},\boldsymbol{x_j}\right\rangle \tag{5}$$

respecting the so-called Karush-Kuhn-Tucker (KKT) conditions equ.6:

$$\alpha_i\geq 0\text{ and }\sum_{i=1}^{n}\alpha_i y_i=0\forall i \tag{6}$$

The solution of the dual problem, with i> 0 only for support vectors x axsis equ.7:

$$\begin{aligned}\boldsymbol{w}&=\sum_s\alpha_s y_s x_s\\ b&=y_k-\sum_s\alpha_s y_s\left\langle\boldsymbol{x}_s,\boldsymbol{x_k}\right\rangle\forall k\text{ s.t. }\alpha_k>0\end{aligned} \tag{7}$$

Eventually the classifying function of new points becomes equ.8:

$$f(x)=\sum_s\alpha_s y_s\left\langle\boldsymbol{x}_s,x\right\rangle \tag{8}$$

Moving to non-linearly separable problems we exploit the so-called soft margin SVM. We allow some points to overcome the margin introducing a slack variable [4]. These variables are regulated by an hyperparameter C representing the trade-off between the necessity of maximizing the margin or minimizing the error. The problem is now as it follows equ.2.3.1:

$$\text{minimize}\,\frac{1}{2}||\boldsymbol{w}||+C\sum_{i=1}^{n}\xi_i$$

with $y_i\left(\langle\boldsymbol{w},\boldsymbol{x_i}\rangle+b\right)\geq 1-\xi_i$ and $\xi_i\geq 0\forall i$

Soft margin SVM alone is not always capable of solving non-linearly separable problems. We exploit the so-called kernel trick: all the points belonging to the input space are remapped in a usually higher dimensional feature space where the problem becomes linearly separable. The trick consists in remapping implicitly all the data without needing to explicitly compute the feature space and perform the transformation of all input data. This is all done by a kernel function defined as equ.9:

$$K(\boldsymbol{x},\boldsymbol{y})=\langle\varphi(\boldsymbol{x}),\varphi(\boldsymbol{y})\rangle \tag{9}$$

being (x) and (y) the transformed of x and y. The kernel function exploited in our project is the Gaussian or Radial Basis Function (RBF) able to transform the input space in an infinite dimensional feature space. The RBF is defined as equ.10:

$$K(x,y)=\exp\left(-\gamma\|x-y\|^2\right) \tag{10}$$

3 ReusIt and Discussion

3.0.1 Evaluating Model Performance

- Confusion Matrix (CM): The confusing matrix, despite its simple logic and structure, is a powerful concept that can provide comprehensive information on how classification works in a variety of research alone. The confusion matrix allows for the calculation of various performance metrics, such as accuracy, precision, recall, and F1 score, which provide insights into the model's performance in terms of correct predictions and errors. By analyzing the values in the confusion matrix, one can assess the model's strengths and weaknesses and understand how well it is performing in different areas of classification

$$\begin{bmatrix}\text{TN} & \text{FP}\\ \text{FN} & \text{TP}\end{bmatrix}$$

- MCC: Matthew Correlation Coefficient is a statistical parameter that produces a high score only if the prediction gives good results. While there is no perfect way of describing the confusion matrix of true and false positives and negatives by a single number, the Matthews correlation coefficient is generally regarded as being one of the best such measures. TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. MCC returns a value between 1 and 1. A coefficient of +1, which represents a perfect prediction, 0 no better than random prediction (wrong prediction), and 1 indicates total disagreement between prediction and observation (completely wrong prediction)equ.11:

$$\text{MCC}=\frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\sqrt{(\text{TP}+\text{FP})\times(\text{TP}+\text{FN})\times(\text{TN}+\text{FP})\times(\text{TN}+\text{FN})}} \tag{11}$$

- TPR or Recall: True positive rate or sensitivity is a statistical parameter that calculates how many of the positive predictions are correctly predicted in real.12:

$$\text{TPR}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{12}$$

- FPR The false positive rate is another statistical parameter that calculates the ratio between the number of negatives that wrongly predicted positive (false positives) equ.13:

$$\text{FPR}=\frac{\text{FP}}{\text{FP}+\text{TN}} \tag{13}$$

- Q2 Score or Accuracy:provides an overall measure of the model's accuracy in correctly classifying instances. It is particularly useful when the dataset is balanced, meaning the number of positive and negative instances is roughly equal. However, in imbalanced datasets, where the number of instances in different classes is significantly different, the Q2 score may not provide an accurate assessment of the model's performance. In such cases, other metrics like precision, recall, F1 score, or area under the ROC curve (AUC-ROC) may be more informative. equ.14:

$$\text{Q2}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}} \tag{14}$$

- Precision:precision represents the proportion of correctly predicted positive instances out of all instances predicted as positive. It focuses on the accuracy of the positive predictions and ignores the instances that were predicted as positive but are actually negative.A high precision value indicates that the model has a low rate of false positives.15:

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{15}$$

- F1 Score:The F1 score combines precision and recall into a single metric that balances both aspects. It ranges from 0 to 1, where a value of 1 indicates perfect precision and recall, and a value of 0 indicates poor performance. It is often used in situations where both precision and recall are important, such as in medical diagnosis or information retrieval systems. It provides a single value that summarizes the model's performance, considering both the ability to make accurate positive predictions (precision) and the ability to capture all positive instances (recall) equ.16:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

3.0.2 Prediction of Signal Peptide in the proteins by SVM method
the result of validation shows that in svm the FN is greater than FP with 74 and 19, respectively. While SVMs generally have good performance, they may struggle to capture certain patterns or features that are important for detecting signal peptides. In your case, SVM had a higher false negative rate (missed actual signal peptides) and lower sensitivity for transmembrane protein prediction. The reason of higher FN than FP in SVM may be is because of Training data bias, because the performance of any machine learning method, including SVM, heavily relies on the quality and representativeness of the training data. If the training data used for SVM is biased or lacks diversity in terms of signal peptide examples, it could lead to a lower sensitivity and higher false negative rate.

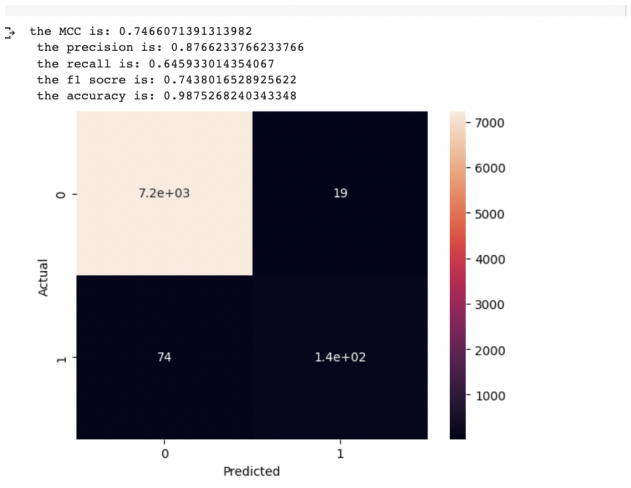

Fig. 16: The result of validation on BM dataset by using SVM training model

Here in the training dataset we had 1723 protein that 258 had signal peptide. It was better that the training dataset was bigger and also had more signal peptide protein in training dataset. The ratio of 1:7 between signal peptide (SP) proteins and overall proteins can be considered biased so in this case, the dataset is skewed towards non-SP proteins, as they are seven times more abundant than SP proteins.(when there is a significant disparity in the distribution of classes, it can introduce a bias towards the majority class and potentially impact the performance of machine learning models). So, the bias towards non-SP proteins can affect the ability of the training model to accurately classify SP proteins. So in this case that both training dataset and benchmark dataset have bias and inbalance, it is very logical that we see higher false negative (FN) rate in SVM compared to false positive (FP) rate. Means, this bias in both dataset resulting in a higher tendency to classify SP proteins as non-SP (FN) rather than misclassifying non-SP proteins as SP (FP). In the other hand, SVM model is biased towards predicting the majority class (non-SP) due to the imbalanced nature of the dataset.

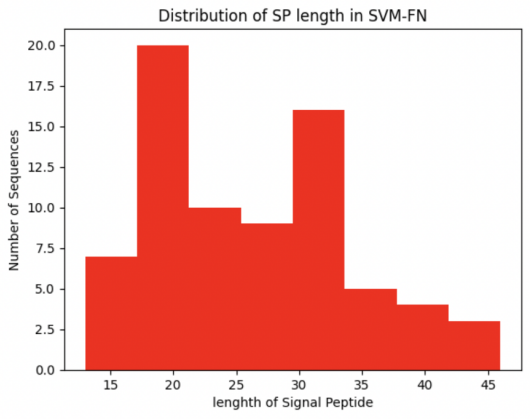

Fig. 17: Distribution of signal peptide length of SVM - False Negative proteins

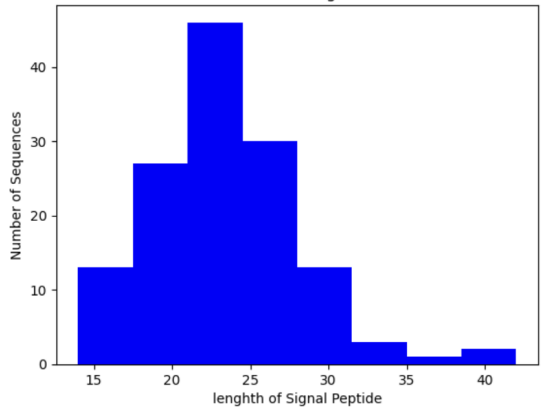

Fig. 18: Distribution of signal peptide length of SVM - True Positive proteins

In the training dataset Figure-,19 the majority (75%) of the data has signal peptides with a length of less than 25 residues. However, among the false negative predictions, 75% ()Figure-17) of the data has signal peptides with a length greater than 20 residues. Additionally, 50% of the false negative predictions have signal peptide lengths greater than 25 residues. we know that distribution of signal peptide lengths in the overall dataset to ensure it is representative of the protein population and this discrepancy suggests that the SVM model may struggle to accurately predict signal peptides with longer lengths. The model might be biased towards shorter signal peptides due to the imbalance in the training dataset, where a larger proportion of the data consists of shorter signal peptides. As a result, the model may be more prone to misclassifying longer signal peptides as false negatives. On the other hand, 75% of the true positives in the signal peptide category had lengths less than 25. This implies that the SVM model is more

successful in identifying shorter signal peptides correctly. Figure-18
One possible explanation for this discrepancy is that the SVM model's decision boundaries or feature representation may be more biased towards shorter signal peptides due to the training dataset's composition. If the majority of the training examples have shorter signal peptides, the model may learn to prioritize and specialize in identifying those patterns, leading to higher accuracy for shorter lengths and lower accuracy for longer lengths.

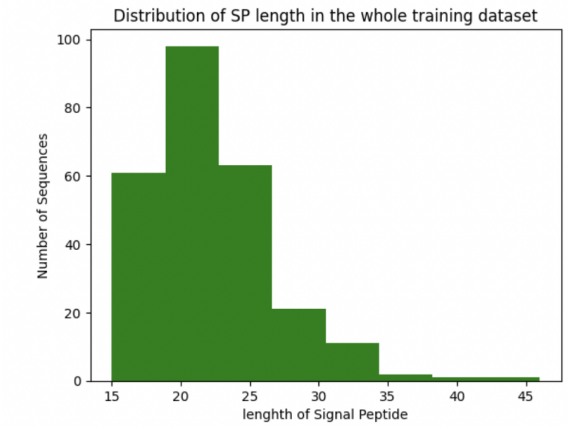

Fig. 19: Distribution of signal peptide length of training dataset

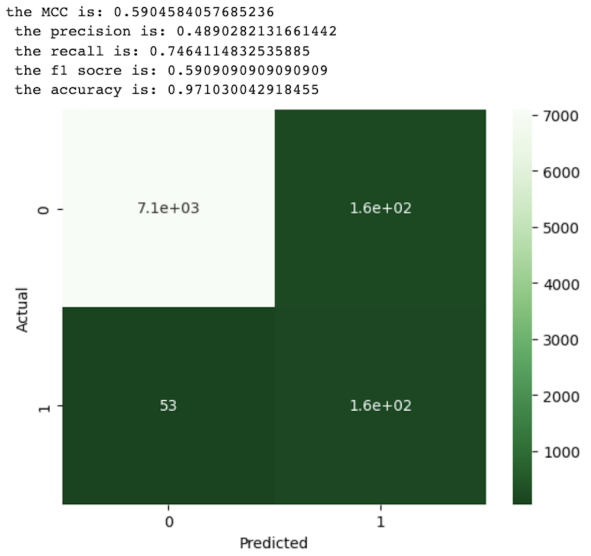

Fig. 20: The result of validation on BM dataset by using VH training model

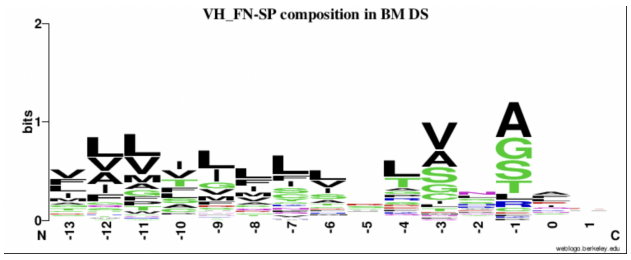

Fig. 21: Motif for log False Negative of VonHeijn method in the Benchmark dataset

3.0.3 Prediction of Signal Peptide in the proteins by VonHeijn method

These values for VonHeijn is higher FP with 163, and FN 53 respectively. This method may have a higher false positive rate, leading to more incorrect predictions. It also has limitations in accurately identifying transmembrane proteins that is because of adopting a completely different modeling approaches. Von Heijn method specifically models the cleavage site on the peptidase part using a window size of 15 amino acids. This approach focuses on the region around the cleavage site, which could provide more discriminative information for signal peptide detection that is very hydrophobic. So if the rate of FPR is higher than FNR in this method, is because that it is struggling to accurately identify transmembrane proteins which has the similar composition of signal peptide around cleavage site (because VonHeijn method modeling the hydrophobicity in a some way); and If we consider the FPR (transmembrane) with the value of .73 , means most of transmembrane proteins that are hydrophobic has detected as signal petide proteins. The reason may be is that VonHeijn model the composition of hydrophobicity of proteins indirectly. In the other side the FPR of Transpeptide proteins is very low (0.04), because this method modeled the trasnpeptid region of proteins , so can capture them very good.

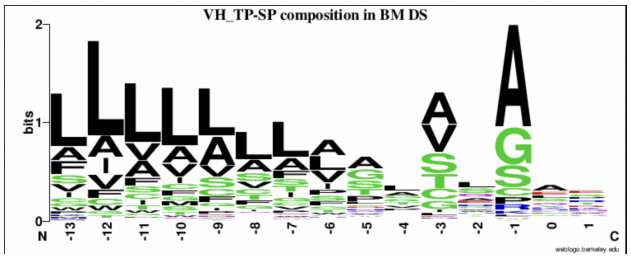

Fig. 22: Motif for log True Positive of VonHeijn method in the Benchmark dataset

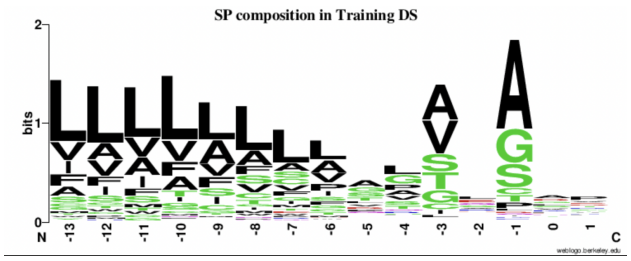

Fig. 23: Motif for log Benchmark dataset

The lower heights of amino acids such as L in the FN logo plot indicate that these residues are less prevalent or less strongly associated with signal peptides in the training data used for the von Heijne method (Figure-21). As a result, the algorithm may have difficulty correctly identifying those instances as signal peptides, leading to FN predictions. There could be multiple reasons for these discrepancies. One possibility is that the algorithm's training data (Figure-23) may not adequately represent the diversity of signal peptides or contain specific examples similar to those present in the FN set. Additionally, the features or patterns used by the algorithm may not fully capture the complexity or variability of signal peptides, leading to misclassifications.

3.0.4 Comparison between the Prediction of VonHeijn and SVM method to detect signal peptides in proteins

To compare the performance of the two methods and determine which one is better for detecting signal peptides, we need to analyze these results and consider their implications.

As you can see in the Table 1 The average MCC for SVM is 0.840, while VH achieves an average MCC of 0.783. This indicates that SVM performs slightly better in terms of overall performance in detecting signal peptides in 5fold-cross-validation.

In terms of Q2, SVM achieves an average value of 0.959, while VH has an average Q2 of 0.947. This suggests that SVM has a higher accuracy in classifying signal peptides compared to VH.

Precision, which measures the proportion of true positives among the instances predicted as positives, is higher for SVM with an average precision of 0.919. VH, on the other hand, has an average precision of 0.863. This indicates that SVM has a higher ability to correctly identify true signal peptides among the predicted positives.

Recall, also known as sensitivity, measures the proportion of true positives correctly identified by the model. SVM achieves an average recall of 0.853, while VH has an average recall of 0.768. This implies that SVM performs better in capturing true signal peptides and minimizing false negatives.

The F1 Score, which combines precision and recall, is higher for SVM with an average value of 0.863, compared to VH with an average F1 Score of 0.812. This suggests that SVM achieves a better balance between precision and recall in detecting signal peptides.

Based on these results, SVM consistently demonstrates higher values across these evaluation metrics, indicating its superior performance in accurately identifying signal peptides.

And finally the obtained result after applying the optimal variable on the trained model based on the Table 2

is that: the MCC (Matthews Correlation Coefficient) is a measure of the overall performance of a binary classification model. It takes into account true positives, true negatives, false positives, and false negatives. The higher the MCC value, the better the model's performance. In this case, the SVM method has a higher MCC value (0.746607) compared to VH (0.590458), indicating better overall performance in detecting signal peptides. The Q2 statistic represents the proportion of correctly classified instances. Again, the SVM method outperforms VH with a Q2 value of 0.987527, suggesting the same accuracy in classifying signal peptides compared to VH (Q2 = 0.971030).Table 2

Precision measures the proportion of true positives among the instances predicted as positives. Precision measures the proportion of true positives among the instances predicted as positives. The SVM method demonstrated a precision of 0.877, indicating a higher ability to correctly identify true signal peptides among the predicted positives. VH had a lower precision value of 0.489, suggesting a higher number of false positives in its predictions. Table 2

Recall, also known as sensitivity, indicates the proportion of true positives correctly identified by the model. The SVM method achieved a higher recall value of 0.746, indicating a better ability to capture true signal peptides and minimize false negatives. VH had a slightly lower recall value of 0.646. Table 2

The F1 Score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. The SVM method achieves a higher F1 Score (0.743802) compared to VH (0.590909), indicating better overall performance in terms of both precision and recall.

Based on these results, the SVM method appears to outperform the VH algorithm in detecting signal peptides. It demonstrates higher values across multiple evaluation metrics, including MCC, Q2, precision, recall, and F1 Score.

Table 1. Average resulted metrics in cross-validation

	HJ	SVM
MCC	0.783018	0.840491
Precision	0.862694	0.918733
Recall	0.767538	0.852923
F1_Score	0.811931	0.863403
Q2	0.946613	0.959380

Table 2. Comparison of VonHeijn and SVM performance on Benchmark dataset

	HJ	SVM
MCC	0.590458	0.746607
Precision	0.489028	0.876623
Recall	0.746411	0.645933
F1_score	0.590909	0.743802
Q2	0.971030	0.987527

4 Conclusion

As the result shown, both methods could detect the signal peptide in a way but the SVM method demonstrates better performance than the VH algorithm in detecting signal peptides based on the evaluation metrics(it shows higher values across multiple evaluation metrics, including MCC, Q2, precision, recall, and F1 Score). To further improve the methods' performance we can consider incorporating additional features, optimizing model parameters, adding some constraints to the model before clustering like domain-specific knowledge into the classification process, or using ensemble methods(Combine multiple classification models, such as SVM with other machine learning algorithms or ensemble techniques like random forests or gradient boosting becasue it can improve the robustness and generalization ability of the models), increasing the training dataset(Expand the training dataset by including more diverse examples

of signal peptides and non-signal peptides with diversity in the signal peptide length. More training data can provide a better representation of the underlying patterns and variations in signal peptides, allowing the models to learn more effectively).

5 References

[1]Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

[2]Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[3]Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[4]Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[5]Gunnar Von Heijne. Patterns of amino acids near signal-sequence cleavage sites. *European journal of biochemistry*, 133(1):17–21, 1983.

[6]Gunnar von Heijne. The signal peptide. *The Journal of membrane biology*, 115:195–201, 1990.