Bioinformatics, 2024, 11 – 18

Build of a model for the detection of signal peptides in proteins

Gianluca Piccolo[1]

[1]International Master Course in Bioinformatics, Alma Mater Studiorum - University of Bologna, Bologna

Email: gianluca.piccolo6@studio.unibo.it

Abstract

Motivation: Signal peptides are among the most common sorting signals, as they target newly synthesized proteins to the secretory pathway. The identification of signal peptides in protein sequences is critical to elucidate protein localization and function. Constructing a model to forecast the occurrence of signal peptides in proteins could offer fresh perspectives on the roles and interactions of proteins with limited experimental data, while potentially uncovering novel drug targets. This study presents two distinct models aimed at this objective: the initial approach utilizes a position-specific weight matrix (von Heijne method), while the second employs Support Vector Machine as a machine learning technique.

Results: Our comparative analysis demonstrates that the Support Vector Machine (SVM from now on) method achieves superior performance with a Matthews Correlation Coefficient (MCC) of 0.81 on the benchmark dataset, compared to 0.62 for the von Heijne method. The SVM approach particularly excels in discriminating between signal peptides and transmembrane regions through the incorporation of multiple sequence features including amino acid composition, hydrophobicity, and charge distribution. Both methods show limitations in handling sequences with non-canonical signal peptide lengths and compositions, suggesting potential areas for future improvement.

Supplementary information: Supplementary materials and data visualization are available on the GitHub repository.

Keywords: Signal peptides, Machine Learning, Support Vector Machines, von Heijne Method, Proteini targeting

1 Introduction

Cells continuously produce proteins with diverse functions. These newly formed proteins must be directed either to various organelles within the cell or transported out of it. To guide this process, nascent proteins possess an inherent Signal Peptide that acts as an "address label" (Chou, 2001). During or after protein translocation, a signal peptidase cleaves the signal peptide at a specific site (Almagro Armenteros et al. 2019).

Secretory signal sequences typically are short sequences of 16 to 30 residues, comprise three structurally and potentially functionally distinct regions: a basic N-terminal region (N-region), a central hydrophobic region (H-region), and a more polar C-terminal region (C-region) (von Heijne, 1983, 1986). The structural elements governing signal sequence cleavage appear to be located in the N- and H-regions, particularly at positions -3 and -1 relative to the cleavage site (von Heijne, 1986). Despite sharing common characteristics, the exact sequences vary considerably among proteins.

The importance of SPs cannot be overstated: they are relevant in the production of recombinant proteins (Mergulhão *et al.*, 2005), a large number of human diseases is caused by mutations in the SPs (Jarjanazi *et al.*, 2007), they also are interesting targets of drugs (Vermeire *et al.*, 2014) and can be exploited as diagnostic biomarkers for several diseases (Dirican *et al.*, 2016).

Given the rapidly expanding number of protein sequences in databases, developing a swift and precise algorithm to identify signal sequences and predict their cleavage sites has become increasingly important (Cai, Lin, and Chou 2003). In the Gene Ontology (GO) framework, the cellular component is one of three aspects describing protein function, alongside biological process and molecular function (Carbon et al. 2021; Ashburner et al. 2000).

Understanding protein localization is crucial for identifying potential protein interactions and surface-exposed targets in drug discovery. To address this need, numerous algorithms have been developed, with SignalP being the pioneer among publicly available methods, now in its sixth iteration. The latest version can detect various types of signal peptides by leveraging a protein language model to represent the motif (Teufel et al. 2022).

In this study, we conducted a similar analysis using two distinct machine learning approaches, both trained on the same dataset. The first method employs a position-specific weight matrix, following the approach of von Heijne (1986). The second utilizes a support vector machine, inspired by the work of Cai, Lin, and Chou (2003).

These complementary approaches aim to provide insights into signal peptide prediction, offering alternative perspectives to the established SignalP algorithm. By comparing these methods, we seek to enhance our understanding of protein targeting mechanisms and contribute to the ongoing development of tools for protein localization prediction.

2 Methods

The data used in these approaches was fetched using the UniProt Knowledgebase release 2024_05 API in Python. To assemble proteins for the datasets, two distinct query searches were executed: one to retrieve proteins belonging to the Positive set (sequences endowed with experimentally determined SP sequences) and another to acquire proteins for the Negative set (sequences lacking an SP).
The datasets included only reviewed entries (UniprotKB) with experimental evidence for the cleavage site. Furthermore, proteins smaller than 40 residues or fragments were excluded, to avoid including fragments.

2.1 Positive and negative datasets

The training dataset consists of 2912 positive eukaryotic sequences (sequences with N-terminal secretory SPs) and 20235 negative eukaryotic sequences (proteins with a subcellular location annotated as cytosolic, nuclear, mitochondrial, plastid, and/or peroxisomal in Eukarya and not belonging to the secretory pathway with experimental evidence).
A data exploratory analysis on the training positive examples has been conducted, showing that the distribution is comparable to the benchmark SP length distribution.
Both methods that will be implemented will exploit the conservation of the average length of SPs in the training and benchmark sets.

The initial dataset underwent rigorous preprocessing to enhance classification accuracy. From the negative dataset, proteins annotated with subcellular localizations in the secretory pathway (endoplasmic reticulum, Golgi apparatus, lysosome) or marked as secreted were excluded to minimize potential misclassification due to the presence of unidentified signal peptides. This filtering step was particularly crucial since these cellular compartments represent typical destinations for SP-containing proteins. The positive dataset was refined by eliminating sequences with signal peptides shorter than 13 residues or those lacking experimental validation of their cleavage sites. This criterion was established based on the minimum length requirement for accurate signal peptide recognition and the need for well-characterized cleavage positions to ensure reliable training data.

To mitigate sequence redundancy, both positive and negative datasets were subjected to clustering analysis using MMSeqs2 (Many-against-Many sequence searching), a software suite designed for sensitive protein sequence similarity search and clustering. The clustering parameters were optimized to maintain biological relevance while reducing redundancy: a minimum sequence identity threshold of 30%, a length coverage threshold of 40%, and a coverage computation mode of 0. This clustering procedure yielded 1,089 representative sequences from the positive dataset and 10,045 representative sequences from the negative dataset. Notably, the resulting ratio of approximately 1:10 (positive:negative) aligns with the optimal class distribution for machine learning approaches in biochemical classification problems, as established by Kurczab et al. This balanced representation helps mitigate potential biases in the subsequent classification tasks while maintaining sufficient diversity in the training data.

2.1.3 Training and Benchmarking set

The training set has been used to train the models, optimized model hyperparameters and perform the cross-validation (CV). Training and benchmarking sets have been introduced splitting each randomized set (positive and negative) with an 80:20 proportion. The subsets containing the 80% of the entries were combined (composing the training set) and the same was made with the 20% entries set (composing benchmarking set). In this way the last two sets have different number of examples, but same proportion of positives and negatives were created.

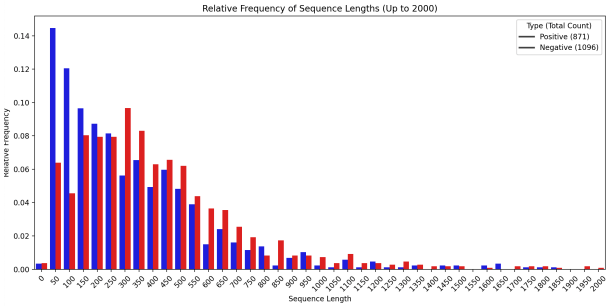

***Figure 1a**: Sequence lengths for Training dataset after the clustering procedure. On x axis the sequence length for proteins in the training set (both positive and negative). On y axis the relative frequencies for specific length.*

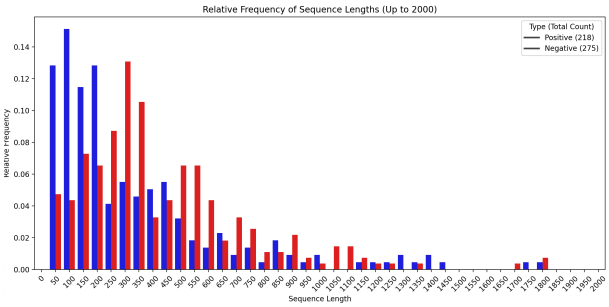

***Figure 1b**: Sequence lengths for Benchmarking dataset after the clustering procedure. On x axis the sequence length for proteins in the training set (both positive and negative). On y axis the relative frequencies for specific length*

2.1.4 Statistical Analysis

Prior to model development, a comprehensive statistical analysis of both training and benchmarking datasets was performed to ensure their suitability for von Heijne and Support Vector Machine (SVM) methodologies. This preliminary analysis was essential for identifying potential biases and validating the datasets' appropriateness for signal peptide detection. The following key aspects were examined:

1. *Signal Peptide Length Distribution*: A comparative analysis of signal peptide lengths between training and benchmarking sets revealed substantial similarity in their distributions (Figure 2a-b). Most signal peptides in both sets exhibited lengths between 20-25 residues, indicating consistency in the

structural characteristics across the datasets.

2. *Amino Acid Compositional Analysis*: Signal peptides demonstrate distinctive amino acid compositions characterized by an elevated proportion of hydrophobic residues. The analysis revealed significant enrichment of hydrophobic amino acids, particularly leucine (L) and alanine (A), compared to the SwissProt background distribution (Figure 3). Notably, both datasets showed a marked decrease in charged amino acid frequencies, consistent with the known physicochemical properties of signal peptides.

3. *Cleavage Site Motif Analysis*: To analyze the cleavage site context, sequences from both datasets were processed to extract regions spanning 13 residues upstream and 2 residues downstream of the cleavage site. Sequence logos generated using WebLogo revealed a conserved pattern across both datasets (Figure 4a,b). The analysis highlighted a characteristic AXA motif near the cleavage site, with a pronounced hydrophobic region dominated by leucine residues in positions -13 to -6.

4. *Taxonomic Distribution*: The kingdom-level taxonomic distribution demonstrated comparable patterns between training and benchmarking sets (fig. 1 supplementary materials).
   The Metazoa kingdom represented the largest proportion in both sets (69.8% and 61.7% respectively), followed by Viridiplantae (14.2% and 19.1%) and Fungi (13.9% and 17.4%). This balanced distribution ensures broad taxonomic representation, crucial for developing robust prediction models applicable across different organisms.

This comprehensive statistical characterization confirms the datasets' compatibility and representativeness, providing a solid foundation for subsequent model development and validation.

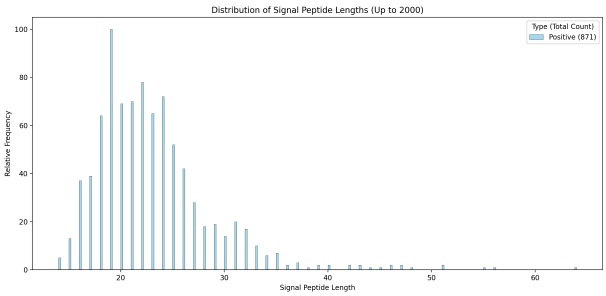

***Figure 2a**. Relative SP frequency in training set.*
Median: 22.00 - Mean: 22.29 - Standard Deviation: 4.30 - Number of sequences after outlier removal: 838 - Percentage of data retained: 96.21%

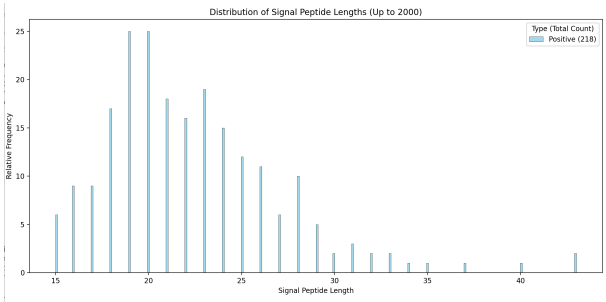

***Figure 2b**. Relative SP frequency in benchmarking set.*
Median: 21.00 - Mean: 22.00 - Standard Deviation: 4.07 - Number of sequences after outlier removal: 213 - Percentage of data retained: 97.71%

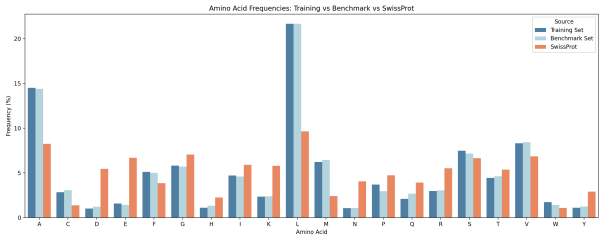

***Figure 3**. Comparison of the aminoacidic composition between training, benchmarking and SwissProt relative frequencies. The frequencies of A and L in training and benchmarking is far more visible than in SwissProts.*

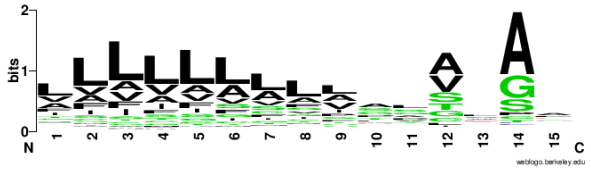

***Figure 4a**. Training set sequence logo. Positions 12-15 show the strongest conservation. The first 12 positions show lower but variable conservation levels. The patterns is reminiscent of a signal peptide cleavage site, where position 12,15 commonly contain small residues like A/G (the "AxA" motif). The pattern indeed matched the von Heijne rules for signal peptide cleavage sites.*

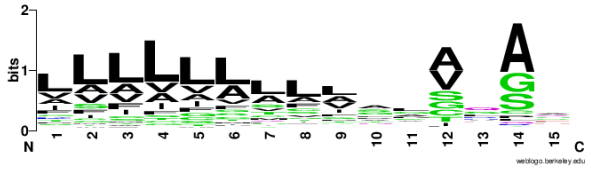

***Figure 4b**. Benchmarking set sequence logo.*

2.1.5 Cross-Validation subsets

The performance evaluation and model optimization were conducted using 5-fold cross-validation, a robust methodology that provides more reliable estimates of model performance compared to single train-test splits. This approach helps mitigate overfitting by evaluating the model's performance across multiple data partitions, offering a comprehensive assessment of its generalization capabilities.

The training dataset was systematically divided while maintaining the original positive-to-negative sample ratio across all partitions. The partitioning process was executed in two phases: first, the training data was separated into positive and negative sets; subsequently, each set was further subdivided into five equal portions. These portions were then paired and merged to create five balanced cross-validation subsets. This stratified approach ensures that each fold maintains a representative distribution of both positive and negative samples, essential for reliable model evaluation and hyperparameter optimization.

The cross-validation procedure not only facilitates the assessment of model stability across different data configurations but also enables the identification of optimal hyperparameters that consistently perform well across various data splits.

2.2 von Heijne Method

2.2.1 Method principles

This method was developed in 1986 by Gunnar von Heijne particularly for the prediction of signal peptide (von Heijne, 1986). This algorithm uses regularized position-specific weight matrices

(PSWM) to represent patterns or motifs present in biological sequences. The number of columns of this matrix is equal to the length of the motifs under study, while the number of rows is represented by the number of characters of the alphabet, that is 4 for nucleotide sequences and 20 for protein sequences.

To compute the PSWM it is first needed to align the N fragments of the sequence motifs of length L. From the aligned fragments a Position-Specific Probability Matrix (PSPM) is computed. This one contains the frequency of each residue k in each position of the motif as in PSWM.

Given a set S of N aligned sequences of length L, where $s_{i,j}$ is the observed residue of aligned sequence i at position j and I is the indicator function ($1\ if\ s_{i,j} = k; 0\ otherwise$), the PSPM M is computed as:

$$M_{k,j} = \frac{1}{N}\sum_{i=1}^{N} I\left(S_{i,j} = k\right) \qquad (1)$$

However, when working with datasets of limited size, certain cells within the PSPM may retain a value of 0. This signifies that a particular residue has not been observed at that specific position during the training process, resulting in a null probability. To circumvent issues arising from 0 values during the subsequent log-odds transformation in the construction of the Position-Specific Weight Matrix (PSWM), pseudocounts are introduced. To address this, the initial values of the PSPM are set to 1 in all cells, assuming the presence of each residue at least once in all positions. As the length of the alphabet is of 20 (amino acids), the PSPM M is calculated as:

$$M_{k,j} = \frac{1}{N+20}\sum_{i=1}^{N} I\left(S_{i,j} = k\right) \qquad (2)$$

The PSWM W is then filled as follows: by computing the log-odds between the frequencies of the PSPM and some background model (like a uniform distribution or the SwissProt composition):

$$W_{k,j} = \log\frac{M_{k,j}}{b_k}$$

where bk is the frequency of residue type k in the background model. This final PSWM can take positive or negative values or zero.

$W_{k,j}$ is a positive number if the frequency of the residue k at position j is higher than the expected frequency given by the background model ($W_{k,j} > b_k$). It means that it is more likely that j could be involved in a functional site, rather than it being a random position, and that it has conserved, because differing from the background model. In the opposite case ($W_{k,j} \leq b_k$) the site is more likely to be a random site, rather than a significant one and $W_{k,j}$is negative or almost equal to zero.

Given any sequence motif or fragment (of length L) $X = [x1, \ldots Xl]$, the score of X's given the computed PSWM W is computed as:

$$score_{(x|w)} = \sum_{i=1}^{n} Wx_i, i \qquad (3)$$

The higher score returned by this procedure is stored and compared to a threshold.

2.2.2 Training and Threshold selection

The von Heijne models were trained using a systematic cross-validation approach to optimize the Position-Specific Weight Matrix (PSWM) and determine the optimal classification threshold. The 5-fold cross-validation process was structured as follows:

Training Phase: Three of the five subsets were utilized to construct the PSWM. For positive sequences within these training subsets, a region spanning from -13 to +2 residues relative to the cleavage site was extracted. These extracted sequences were used to compute position-specific amino acid frequencies, which were then normalized against SwissProt background frequencies to generate the PSWM.

Validation Phase: One subset was designated for threshold optimization. The following procedure was implemented:

1. The first 90 N-terminal residues were extracted from both positive and negative sequences.
2. A sliding window approach was employed, where:
    - Window size: 15 residue
    - Window positions: 1-15, 2-16, 3-17, ..., 76-90
    - Each window position generated a log-likelihood score using the PSWM
3. For each sequence, the maximum score across all window positions was designated as the global score, reflecting the likelihood of signal peptide presence.
4. Threshold optimization was performed by:
    - Computing precision and recall values across various threshold points
    - Generating a precision-recall curve
    - Calculating F1 scores for each precision-recall pair
    - Selecting the threshold that maximized the F1 score

Testing Phase: The remaining subset was used for model evaluation. The procedure involves:

1. Extracting the first 90 N-terminal residues from test sequences
2. Applying the same sliding window approach to compute global scores
3. Classifying sequences based on the optimized threshold:
    - Scores above threshold → Positive prediction (signal peptide present)
    - Scores below threshold → Negative prediction (signal peptide absent)

This entire process was repeated five times, with each subset serving once as the validation set and once as the test set, while maintaining different combinations of training subsets. This rotation ensured robust evaluation of the model's performance and stability. The final threshold for benchmark evaluation was determined by averaging the optimal thresholds from all five iterations.

2.2.3 Prediction

For the final model evaluation on the benchmark dataset, all positive sequences from the training set (encompassing all five cross-validation subsets) were utilized to construct the Position-Specific Weight Matrix (PSWM). Prediction scores were computed using the same sliding window methodology established during the training phase. The classification threshold was set to 6.18, derived from averaging the optimal thresholds identified during the five-fold cross-validation process (table 1). This composite model was then applied to classify sequences in the benchmark dataset,

and its performance was evaluated using standard metrics.

2.3 Support Vector Machine Method

2.3.1 Method principles

Support Vector Machine (SVM) is a supervised learning algorithm that performs binary classification by determining an optimal separating hyperplane between classes in a feature space. The algorithm seeks to maximize the margin—the distance between the hyperplane and the nearest data points from each class, known as support vectors. When the data is not linearly separable in the input space, SVM employs the "kernel trick," which implicitly maps the input data into a higher-dimensional feature space where linear separation becomes feasible. This transformation is achieved through kernel functions that compute inner products in the feature space without explicitly calculating the mapping, making the algorithm computationally efficient. The effectiveness of SVM lies in its ability to find a decision boundary that maximizes generalization performance while minimizing structural complexity.

Given a binary classification case, we can define

$$D = \{(x_i, y_i) i = 1,2, \dots, n\}$$

As a training set D comprises n-points in the d-dimensional space $x_i \in \mathbb{R}^d$

$$y_i \in \{-1, 1\}$$

As the vector that contains the classes.

So, the linear separator has the equation:

$$w^T x_i + b = 0$$

A Support Vector (SV from now on) is a point belonging to a class that is the closest to this linear separator, which is a line in $\mathbb{R}^2$ and a hyperplane in $\mathbb{R}^n$.

The distance between an SV to the linear separator can be computed as:

$$d = \frac{1}{||w||}$$

It is possible to define the margin ρ as $2 * d$.

The main aim of SVM is to maximize the margin (or minimize $\frac{1}{2}||w||^2$) under the constraint of $y_i(w^T x_i + b) \geq 1$.

The resolution of this problem can be solved introducing the Dual Lagrangian function, thanks to which a multiplier is associated to every point of each class. In particular:

$$L(\alpha_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \qquad (4)$$

$$for\ \alpha_i \geq 0, where \sum_{i=1}^{n} \alpha_i y_i = 0$$

There are some cases in which points belonging to different classes cannot be linearly separable. To manage these situations, SVM can allow for error (introducing a slack variable ξ) in classification (SVM with soft margin) or it is possible to introduce Kernels.

2.3.2 Kernels

When points of the training set in original space are not linearly separable, the kernel trick is introduced. This method allows mapping the points in the original space into a new space (feature space) in a different dimension, allowing a non-linear transformation of the data. This is done thanks to a transformation:

$$\phi(x_i) = \phi(x)$$

A Kernel function is defined as:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \qquad (5)$$

it computes the dot product between transformed vectors without explicitly performing the transformation itself. Thanks to the Kernel Trick, the dual Lagrangian becomes:

$$L(\alpha_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j x_j \langle \phi(x_i), \phi(x_j) \rangle$$

Among the different type of Kernels, for this project the chosen one was the Radial Basis Function (RBF) (Cai et al., 2003), defined as follow:

$$K(x_i, x_j) = \exp\left(- \frac{\left|\left| x_i - x_j \right|\right|^2}{2\sigma^2} \right) \qquad (6)$$

2.3.3 *SVM Application for SP prediction*

The data points – representing query proteins in this context – are transformed into an N-dimensional space, where N corresponds to the number of selected features for evaluation. Evaluating the significance of each feature becomes crucial in understanding the relationship within the data.
While the von Heijne method relies solely on residue composition, SVM can be trained on a wider array of potential properties.
The combination of different input features and hyperparameters tuning generate different trained models (grid search). The best model sued for the final prediction has been retrieved with a cross-validation approach.

2.3.4 *Feature* selection *and encoding*

Based on the established characteristics of signal peptides (von Heijne, 1990), four primary features were selected for analysis:

1. *Amino Acid Composition*: Signal peptides exhibit a distinct amino acid distribution compared to the background SwissProt composition, making this characteristic a valuable discriminative feature.

2. *Hydrophobicity*: To capture the characteristic hydrophobic core of signal peptides, the Kyte & Doolittle hydropathicity scale was employed].

3. *Charge Distribution*: To represent the positively charged N-terminal region, a binary charge scale was implemented (1 for charged amino acids, 0 for uncharged).

4. *α-Helix Propensity*: The Chou & Fasman conformational parameters [14] were utilized to quantify the tendency of the

hydrophobic core to form α-helical structures.

The amino acid composition served as the baseline feature across all models. Each protein sequence was encoded as a 20-dimensional vector, representing the frequency of each amino acid within the first k N-terminal residues, where k was treated as a hyperparameter. The value of k was constrained to five possible values based on the observation that most signal peptides range from 20 to 24 residues in length.

For the remaining features, a sliding window approach was implemented to compute localized average values. Window sizes of 5 residues were used for hydrophobicity and charge calculations, while a 7-residue window was employed for α-helix propensity measurements. For each feature, three specific values were incorporated into the protein's feature vector:

- The global average across all windows
- The maximum value observed
- The normalized position of the maximum value within the first k residues

The expected positioning of these features aligns with the structural organization of signal peptides: maximum charge values typically occur near the N-terminus in the charged region, while peak hydrophobicity and α-helix propensity values are generally found toward the C-terminal end, corresponding to the hydrophobic core. To ensure uniform scaling across features, all values were normalized to the range [0,1] using Min-Max scaling.

2.3.5 Hyperparameter

This configuration requires the optimization of two critical hyperparameters:

1. *Gamma (γ):* This parameter governs the decision boundary's geometry by controlling the radius of influence of individual training examples. A lower γ value results in a broader influence zone, creating a more generalized decision surface that accommodates greater data variation. Conversely, higher γ values restrict each data point's influence to its immediate vicinity, potentially leading to a more complex decision boundary that may be susceptible to overfitting.

2. *Regularization Parameter (C):* This parameter mediates the balance between decision boundary smoothness and classification accuracy. Lower C values permit more flexible decision boundaries, potentially improving generalization on complex datasets while risking underfitting. Higher C values enforce stricter classification constraints, which may enhance accuracy on training data but could compromise generalization performance.

The hyperparameter optimization was conducted through an exhaustive grid search across the following parameter space:

- Regularization parameter (C): {1, 2, 4, 8}
- Gamma (γ): {0.5, 1, 2, "scale"}

A five-fold cross-validation protocol was implemented to identify optimal hyperparameter combinations, with the dataset partitioned as follows:

1. Training Set (60%): Three folds were utilized to train 80 distinct models, each representing a unique combination of hyperparameters and feature encodings.

2. Validation Set (20%): One-fold was employed to evaluate model performance and identify optimal hyperparameter combinations based on Matthews Correlation Coefficient (MCC) maximization.

3. Testing Set (20%): The remaining fold was reserved for unbiased performance assessment of the optimally parameterized model.

This cross-validation procedure was executed five times for each of the eight feature combination configurations, ensuring robust hyperparameter selection and performance estimation.

2.3.5 Model *selection*

The optimal model configuration for each feature combination was determined through a systematic aggregation of cross-validation results. For each feature set, the performance metrics from the five cross-validation iterations were averaged, with hyperparameters selected based on their modal frequency across iterations. This process yielded eight candidate models, from which the final model was selected based on the highest mean Matthews Correlation Coefficient (MCC).

In cases where multiple hyperparameter combinations achieved equal frequency during cross-validation, a hierarchical selection criterion was applied:

1. For regularization parameter C and kernel coefficient γ, the lower values were preferred to minimize model complexity.

2. For sequence length parameter k, the value nearest to the median (22) was selected, aligning with the empirically observed distribution of signal peptide lengths.

The final model, incorporating the optimal feature combination and hyperparameter values, was subsequently trained on the complete training dataset, encompassing all five cross-validation subsets. This comprehensively trained model was then employed to evaluate signal peptide presence in the independent benchmark dataset, utilizing identical feature encoding procedures as applied during training to ensure consistency in prediction.

2.4 Performance measurements

Both SP detection methods performances were assessed started from the confusion matrix. In particular, the following metrics have been computed:

- Matthews Correlation Coefficient (MCC) is a quality measure for classification, particularly useful in case of imbalanced datasets.
  MCC ranges from -1 (model prediction always wrong) to +1 (observations coincide with prediction). If a MCC is equal to 0, then the model is a random classifier. MCC is computed as follow:

$$MCC = \frac{TPxTN - FPxFN}{\sqrt{(TP + FP)x(TP + FN)x(TN + FP)x(TN + FN)}}$$

- Precision is a measure that signifies the accuracy of predicted positive instances by determining the proportion of true positives among them.

$$Precision = \frac{TP}{TP + FP}$$

- Recall is a metric that assesses the classifier's ability to correctly identify actual positive instances, indicating the proportion of true positives among the total positive instances.

$$Recall = \frac{TP}{TP + FN}$$

- Accuracy is a metric that calculates the proportion of corrected predictions relative to the total number of predictions made. It is formed by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- F1 score represents the harmonic mean of precision and recall, exhibiting elevated values under specific combinations of precision and recall. Formally, the F1 scores is defined as:

$$F1 = \frac{Recall\ x\ Precision\ x\ 2}{Precision + Recall}$$

3 Results

3.1 *Cross*-validation results

Different methods have been analyzed on their performance.

For von Heijne approach, the average threshold of its 5 cross-validation process has been taken to account in the prediction of the presence of SPs.

Cross Validation	Threshold
1	7.60
2	6.80
3	5.80
4	5.60
5	5.10
Average	6.18

***Table 1**: Cross-validation thresholds for the von Heijne method. The table shows the threshold values obtained for each fold during the 5-fold cross-validation process and their arithmetic mean. These thresholds represent the optimal cutoff values for classifying sequences as containing signal peptides based on the Position-Specific Weight Matrix scores. The decreasing trend in threshold values across folds reflects the adaptation of the model to different subsets of the training data, while maintaining robust classification performance. The average threshold of 6.18 was used for final predictions on the benchmark dataset.*

For the SVM method, hyperparameter optimization was performed through a five-fold cross-validation procedure. The optimal hyperparameters were determined by identifying the most frequently occurring values that maximized the Matthews Correlation Coefficient (MCC) across all cross-validation folds for each feature combination evaluated. This systematic approach to parameter selection helps ensure the model's robustness while mitigating potential overfitting by validating performance across multiple data partitions.

Cross Validation	γ	C
1	0.5	8
2	0.5	8
3	0.5	8
4	0.5	8
5	0.5	8

***Table 2.** Optimized hyperparameters for the Support Vector Machine model across 5-fold cross-validation. The table shows the consistent selection of γ (gamma) = 0.5 and C = 8 as optimal parameters across all folds. The gamma parameter defines the influence radius of each support vector, while C controls the trade-off between margin maximization and training error minimization. The consistency of these values across all folds suggests a stable model configuration that generalizes well to different subsets of the training data. More results analysis on supplementary materials on GitHub repository.*

3.2 Benchmarking results

For evaluating predictions on the benchmark set, both methods underwent distinct preparation approaches. The von Heijne method's PSWM matrix was constructed using all positive sequences from the training dataset. Similarly, for the SVM approach, the model was trained on the complete training set using the optimal hyperparameters (γ = 0.5, C = 8) identified during cross-validation. As shown in Table 4, the SVM method demonstrated markedly superior performance compared to the von Heijne method, maintaining consistency with the cross-validation results. This improvement is particularly evident in the MCC score, where the SVM method showed a substantial increase from 0.635 (von Heijne) to 0.819 (SVM), highlighting its enhanced predictive capability.

Metric	Von Heijne	Support Vector Machine
MCC	0.635 ± 0.027	0.819 ± 0.011
Precision	0.795 ± 0.023	0.882 ± 0.015
Recall	0.807 ± 0.024	0.920 ± 0.013
Accuracy	0.819 ± 0.025	0.910 ± 0.012
F1	0.788 ± 0.022	0.891 ± 0.014

***Table 3**: Performance comparison between von Heijne and Support Vector Machine methods during cross-validation. The table presents five standard classification metrics (MCC: Matthews Correlation Coefficient, Precision, Recall, Accuracy, and F1 score) with their respective mean values and standard errors across the 5-fold cross-validation. The SVM consistently outperforms the von Heijne method across all metrics, with notably higher MCC (0.819 vs 0.635) and Recall (0.920 vs 0.807). The lower standard errors in the SVM results also indicate more stable performance across different data splits.*

As hypothesized, the SVM approach demonstrated superior performance compared to the von Heijne method, as evidenced by the higher MCC values (Table 3). This improvement can be attributed to the SVM's ability to capture complex, non-linear relationships in the feature space through its kernel function and optimal hyperparameter configuration.

These results underscore the critical role of optimization procedures in determining optimal thresholds and hyperparameters for both methods. Although both approaches exhibited misclassification of positive and negative proteins (as detailed in Table 4), the SVM method demonstrated notably lower error rates. This superior performance of the SVM approach can be attributed to the inherent advantages of machine learning methods, which typically offer greater robustness, scalability, and flexibility compared to algorithmic approaches. The SVM's capability to incorporate multiple discriminative features during training - including amino acid composition, hydrophobicity patterns, and charge distribution - provides it with a significant advantage over the von Heijne method, which is primarily limited to analyzing positional amino acid frequencies around the cleavage site. This fundamental difference in feature utilization explains the SVM's enhanced ability to handle the diverse compositional patterns found in signal peptides.

Method	TP	TN	FP	FN	FPR	FNR
vH	176	228	47	42	17.1%	19.3%
SVM	196	248	27	22	9.82%	10.1%

***Table 4**: Confusion matrix metrics for von Heijne (vH) and Support Vector Machine (SVM) methods on the benchmark dataset. The table shows True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), False Positive Rate (FPR), and False Negative Rate (FNR) for both methods. The SVM achieves better performance with higher TP (196 vs 176) and TN (248 vs 228) counts, while maintaining lower error rates in both false predictions (FPR: 9.82% vs 17.1%; FNR: 10.1% vs 19.3%). This indicates SVM's superior ability to correctly identify both the presence and absence of signal peptides.*

3.3 False negative and false positive

Error analysis revealed notable differences in the performance of both methods. The von Heijne approach resulted in 47 false positives and 42 false negatives out of 493 predictions, yielding a false positive rate (FPR) of 17.1% and a false negative rate (FNR) of 19.3%. In contrast, the SVM method demonstrated superior accuracy with only 27 false positives and 22 false negatives, corresponding to a significantly lower FPR of 9.82% and FNR of 10.1%. The primary source of misclassification for both methods was the presence of transmembrane domains in the N-terminal region. In the von Heijne method, 83% of false positives (39 out of 47) contained transmembrane domains, with 23.21% (39/168) of all sequences containing transmembrane regions being incorrectly classified. The SVM approach showed improvement in this aspect, with 77.78% of false positives (21 out of 27) containing transmembrane domains, and only 12.5% (21/168) of transmembrane-containing sequences being misclassified. These results suggest that while both methods struggle with distinguishing between signal peptides and transmembrane domains due to their similar physicochemical properties, the SVM's incorporation of transmembrane propensity features and additional sequence characteristics provides a more robust approach to this classification challenge. The reduced error rates in the SVM method, particularly in handling transmembrane regions, demonstrate its enhanced ability to discriminate between these similar structural elements.

4 Conclusions

In this study, we developed and compared two computational approaches for signal peptide (SP) prediction in protein sequences: a position-specific weight matrix (PSWM) method based on von Heijne's algorithm and a Support Vector Machine (SVM) classifier. The dataset was divided into training and benchmarking sets, with the training set further partitioned into five equal subsets for cross-validation purposes.

The von Heijne method was implemented by constructing a PSWM from the cleavage site regions of SP-containing proteins in the training set. A classification threshold was established by averaging the optimal thresholds determined during cross-validation. Sequences with scores above this threshold were classified as containing SPs, while those below were classified as non-SP sequences. For the SVM approach, we evaluated eight distinct models incorporating various combinations of key SP characteristics: hydrophobicity, amino acid charge, α-helix propensity, and amino acid composition. Model selection was based on Matthews Correlation Coefficient (MCC) performance, with the optimal model utilizing amino acid composition, hydrophobicity, and charge features.

The SVM classifier demonstrated superior performance compared to the von Heijne method across all evaluation metrics. Analysis of misclassifications revealed that false positives (FPs) in both methods were primarily associated with the presence of transmembrane domains and, to a lesser extent, transit peptides. This can be attributed to the hydrophobic regions present in these sequences, which share characteristics with SP hydrophobic cores.

False negative (FN) analysis revealed distinct patterns for each method. In the von Heijne approach, misclassified sequences showed reduced conservation of the hydrophobic core near the cleavage site compared to the training sequences. The SVM method's FNs exhibited both compositional variations in the encoded residues and atypical SP lengths, with a broader distribution of both shorter and longer sequences compared to the benchmark set.

Future improvements in SP prediction could involve developing ensemble SVM models with multiple kernels that account for SP structural features. Additionally, deep learning approaches show promise for enhancing both SP detection and cleavage site identification, as demonstrated in recent literature. These advanced techniques could potentially capture more complex sequence patterns and relationships than traditional methods.

References

Almagro Armenteros,J.J. et al. (2019) SignalP 5.0 improves signal peptide predictions using deep neural networks. Nature Biotechnology, 37, 420-423.

Ashburner,M. et al. (2000) Gene Ontology: tool for the unification of biology. Nature Genetics, 25, 25-29.

Cai,Y.D., Lin,S.L., and Chou,K.C. (2003) Support vector machines for prediction of protein signal sequences and their cleavage sites. Peptides, 24, 159-161.

Carbon,S. et al. (2021) The Gene Ontology Resource: enriching a GOld mine. Nucleic Acids Research, 49, D325-D334.

Chou,K.C. (2001) Prediction of signal peptides using scaled window. Peptides, 22, 1973-1979.

Dirican,N. et al. (2016) The diagnostic significance of signal peptide-complement C1r/C1s, Uegf, and Bmp1-epidermal growth factor domain-containing protein-1 levels in pulmonary embolism. Annals of Thoracic Medicine, 11, 277-282.

Jarjanazi,H. et al. (2007) Biological implications of SNPs in signal peptide domains of human proteins. Proteins: Structure, Function, and Bioinformatics, 70, 394-403.

Mergulhão,F.J.M. et al. (2005) Recombinant protein secretion in Escherichia coli. Biotechnology Advances, 23, 177-202.

Nielsen,H., Tsirigos,K.D., Brunak,S., and von Heijne,G. (2019) A brief history of protein sorting prediction. The Protein Journal, 38, 200-216.

Teufel,F. et al. (2022) SignalP 6.0 predicts all five types of signal peptides using protein language models. Nature Biotechnology, 40, 1023-1025.

Vermeire,K. et al. (2014) Signal peptide-binding drug as a selective inhibitor of co-translational protein translocation. PLoS Biology, 12, e1002011.

von Heijne,G. (1983) Patterns of amino acids near signal-sequence cleavage sites. European Journal of Biochemistry, 133, 17-21.

von Heijne,G. (1986) A new method for predicting signal sequence cleavage sites. Nucleic Acids Research, 14, 4683-4690.

von Heijne,G. (1990) The signal peptide. The Journal of Membrane Biology, 115, 195-201.