Sequence analysis

Comparison of Von Heijne and SVM model for classification Signal Peptides (SP)

Krsto Vujovic1, *

[1]Department of Pharmacy and Biotechnology, Alma Mater Studiorum Università di Bologna.

Abstract

Motivation: Comparing two algorithms (methods) for Signal Peptide (SP) prediction. The aim of this project is to see which approach performs better, Von Heijne method or Support Vector Machine (SVM). Given that Von Heijne is statistical model and SVM is machine learning technique, expectations are that machine learning method will be better.

Results: As expected, results showed that SVM (machine learning approach) does a better job at predicting signal peptide sequence. In fact, in all five metrics measured, it managed to outperform Von Heijne. But both methods had problems with false negative and false positive cases.

Contact: Krsto.vujovic@studio.unibo.it

Supplementary information: Supplementary data are available at GitHub link.

1 Introduction

Subcellular protein sorting, i.e. the processes through which proteins are routed to their proper destination within a cell, is a fundamental aspect of cellular life. In many cases, sorting depends on `signals' that can already be identified by looking at the primary structure of a protein[1].

In both prokaryotic and eukaryotic cells, proteins are allowed entry into the secretory pathway only if they are endowed with a specific targeting signal: the Signal Peptide (SP)[2]. SPs are short N-terminal amino acid sequences that target proteins to the secretory (Sec) pathway in eukaryotes and for translocation across the plasma (inner) membrane in prokaryotes[3].

Signal peptides are cleaved by one of a small class of enzymes known as signal peptidases once its targeting function has been carried out[2].

Early on, comparisons of known SPs indicated that they typically have three distinct domains (Fig. 1): an amino-terminal positively charged region (N-region, 1-5 residues long); a central, hydrophobic part (H-region, 7-15 residues); and a more polar carboxy-terminal domain (C-region, 3-7 residues). Beyond this overall pattern, no precise sequence conservation could be found, and it soon became obvious that SPs are highly variable, rapidly evolving structures[2].

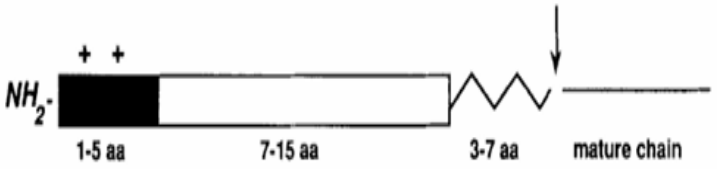

Figure 1. The basic design of signal peptide[2].

The identification of signal peptides in protein sequences is an important step toward protein localization and function characterization[4]. While prediction of sorting signals (SP) has a long history, started by the early work on secretory signal peptides (von Heijne, 1983; McGeoch, 1985; von Heijne, 1986b), it is only with the application of modern machine learning techniques, such as neural networks (NNs) and hidden Markov

models (HMMs), that seems to be approaching the necessary levels of accuracy[1].
At the top of the hierarchy of state of the art models, we have models such as SignalP 6.0, SignalP 5.0, DeepSig, and LipoP.

DeepSig is a novel approach to predict signal peptides in proteins based on deep learning (with Deep Convolutional Neural Network (DCNN) architecture), it is specifically tuned to recognize signal peptide sequences and sequence labelling methods[4].

The most recent advancement in SP prediction is SignalP 6.0, a method based on protein language models (LMs) able to detect all SP types and to extrapolate information from distantly related proteins and metagenomic data, differently from its predecessor, SignalP 5.0[3].

The aim of this project is, to present a comparison between two models, Von Heijne and Support Vector Machine (SVM), for the prediction of signal peptide sequences on unseen proteins. Von Heijne is one of the first available models for this purpose, whereas SVM is a modern machine learning method that has gained significant popularity with the expansion of neural networks (NN).
As discussed in the results and discussion section, the SVM outperforms the Von Heijne model, which is expected, given that it is a more sophisticated model.

2 Methods

2.1 Datasets

Sequences were sourced from the UniProtKB/SwissProt database (release 2023_04)[5]. The positive dataset consists of reviewed eukaryotic entries longer than 30 residues, with experimental evidence confirming the presence of signal peptides (ECO:0000269).

The negative dataset shares some filters with the positive set (only eukaryotic proteins longer than 30 residues) but excludes sequences with signal peptides at any evidence level. Instead, it includes proteins experimentally verified to be localized in cellular compartments such as the cell membrane, cytosol, mitochondrion, nucleus, peroxisome, and plastid, where signal peptides should not be found.

The exclusion of sequences shorter than 30 residues avoids the inclusion of fragments. The raw datasets underwent further processing.

For the negative set, entries with keywords "lysosome," "secreted," "endoplasmic," and "Golgi" were removed, as these are locations where signal peptide containing proteins are typically transported.

From the positive set, signal peptides shorter than 13 residues or with unknown cleavage sites were filtered out. After this initial filtering, the positive set was reduced to 2,937 sequences, and the negative set to 30,011 sequences.

To address redundancy, which can bias the model's learning process and increase computational load, both datasets were clustered using MMseqs2 (Many-against-Many sequence searching)[6] with a 30% sequence identity threshold and a 40% length coverage threshold on both query and target (cov-mode 0).

This clustering resulted in a positive dataset with 1,093 representative sequences and a negative dataset with 9,358 representative sequences.

Using UniProt's ID mapping tool, comprehensive metadata were retrieved for statistical analysis, including fields such as Entry ID, SP Cleavage Site position (for positive proteins), protein length, taxonomic lineage, organism, and sequence.

The datasets were then divided into training and benchmarking sets by randomly allocating 80% of the sequences to the training set and the remaining 20% to the benchmarking set. The training set was further split into five subsets for 5-fold cross-validation, optimizing the model parameters before testing on the benchmarking set.

The final training set comprised 8,360 sequences, while the benchmarking set contained 2,091 entries. The entire filtering process was executed using bash and Pandas library.

To make sure that dataset is prepared correctly and ready for modelling it is crucial to provide a statistical analysis of the training and benchmarking datasets.
Starting with plot for distribution of SP length in both benchmark and training set, Figure 2.

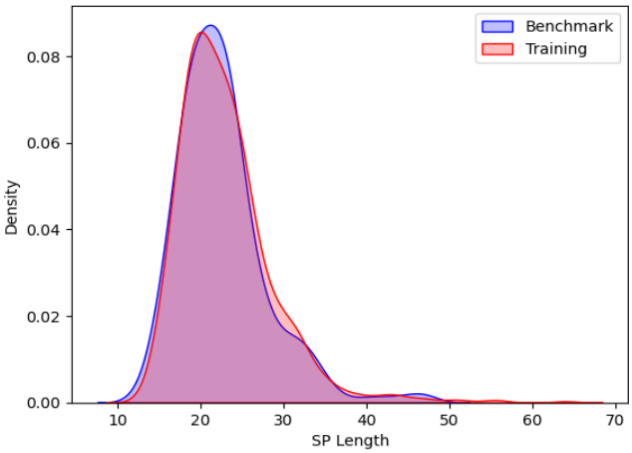

Figure 2. Density distribution of SP length (benchmark vs training)

As it can be seen from the plot, density shape is same for both sets, there is no significant difference between two sets, actually median for SP in both set is exactly the same 22 residues.
It's very important to point out, that should we find significant difference in statistical analysis, it can affect our performance and whole process of data preprocessing because we would have to repeat the process and find better way for generating datasets.

Signal peptides are distinguished not only by their length, ranging from 15 to 30 residues, but also by their unique amino acid composition. Specifically, these sequences tend to have a higher proportion of hydrophobic amino acids, such as alanine (A), leucine (L), and valine (V), which are essential for interacting with the translocation machinery, these sequences also tend to have lower proportion of polar charged amino acids like lysine (K), glutamic acid (E), aspartic acid (D), arginine (R).
Figure 3 clearly demonstrates the differences in amino acid composition between the training and benchmarking sets in comparison to the composition that we downloaded from the SwissProt[7] as our baseline.

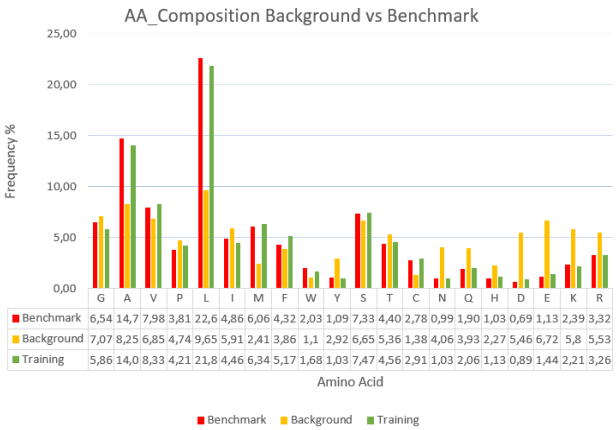

Figure 3. Comparison of amino-acid composition between Benchmark set, Training set and SwissProt composition (background).

As it can be seen on Figure (3) there is no statistical difference between Training and Benchmark set, and clearly there are patterns that can pointed out when we compare those two to the baseline distribution, as it should be because composition of SP has some distinct features.

Third plot is very important step for our datasets analysis. It is the evaluation of motifs in the cleavage site context. The positive sequences from both sets have been processed in order to retain only 13 residues upstream and 2 residues downstream the cleavage site position.

WebLogo generates sequence logos, graphical representations of the patterns within a multiple sequence alignment. Sequence logos provide a richer and more precise description of sequence similarity than consensus sequences and can rapidly reveal significant features of the alignment otherwise difficult to perceive[8].
The overall height of each stack indicates the sequence conservation at that position (measured in bits), whereas the height of symbols within the stack reflects the relative frequency of the corresponding amino or nucleic acid at that position[8].

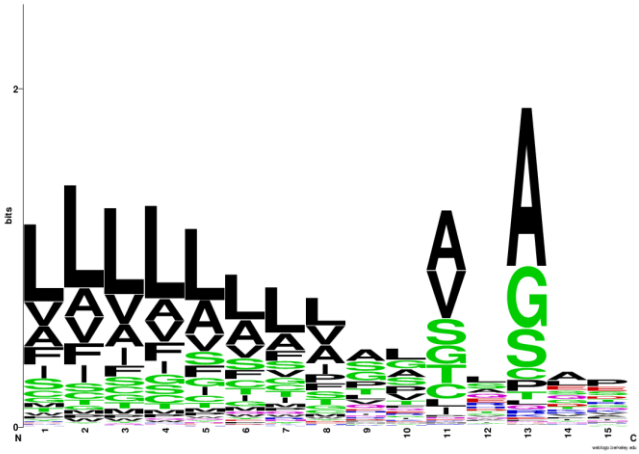

Figure 4. Sequence logo of Training set, as said above first thirteen residues are upstream (-13) and then last two residues are upstream (+2) from cleavage site.

The hydrophobic chain that corresponds to the H-region of SP sequence is clearly visible from position 1 to position 6-7, with the overrepresentation of apolar residues such as leucine, followed by valine and alanine. On the other hand C region (3-7 amino acids) corresponds to positions from 7^{th} till 13^{th}, the cleavage site motif is [A,V]XA.

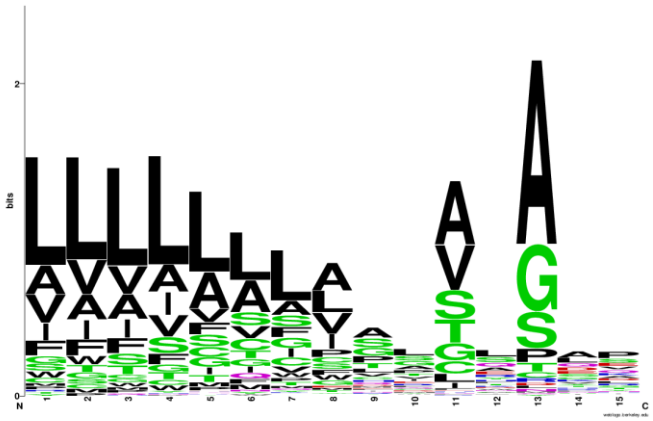

Figure 5. Sequence logo of Testing set.

There are only minor differences in comparison to training set, so first couple of leucine are higher in their representation but pattern is same from 1^{st} till 6^{th} position, and then cleavage site motif [A,V]XA. All other plots can be found in supplementary material.

2.2 Von Heijne model

The von Heijne method is a significant milestone in signal peptide prediction, developed in 1986 by Gunnar von Heijne. This computational and statistical approach utilizes a Position Specific Weight Matrix (PSWM) with dimensions of 20x15, representing the alphabet length and motif length, respectively. The motif focuses on the cleavage site context, spanning 13 residues upstream to 2 residues downstream of the site, creating a 15-residue window centered around the cleavage site.

The PSWM captures the likelihood of specific residues at each position within the motif. To compute this, a Position Specific Probability Matrix (PSPM) with the same dimensions is required. Given a set S of N sequences of length L, the frequency M of residue k at position j within the motif can be determined (Formula 1).

$$M_{k,j} = \frac{1}{N}\sum_{i=1}^{N} I(s_{i,j} = k)$$

Formula 1. Filling of PSPM

To explain the formula presented above:
$s_{i,j}$ is the observed residue of aligned sequence i at position j; k is the residue corresponding to the k-th row in the matrix; $I(s_{i,j}= k)$ is an indicator function (1 if the condition is met, 0 otherwise).
Now since we are dealing with databases of finite size there is one potential problem with PSPM designed like this. Some cells of the PSPM could remain 0, meaning such residue has never been observed in that precise position during the training procedure and thus the probability is null. Also, since later computation of PSWM starts from the PSPM and doing log-odd transformation, zero values are something that we have to avoid.
For this reason, we introduced the concept of pseudo-counts where PSPM is initialized with 1 in all cells, assuming each residue is observed at least once in all positions. This modification also changes the N, number of the sequences. Therefore, a need to modify formula accordingly.

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

Formula 2. Filling PSPM with pseudo counts implementation.

Therefore, formula have same structure, but it is initialized at 1 and averaged over N + 20 (number of sequences plus pseudo counts).

Now this PSPM serves as base for computation of PSWM, where W is computed using composition of SwissProt as background (b_k). The equation is the following formula (Formula 3):

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

Formula 3. Filling of PSWM from PSPM.

Therefore, to explain meaning of Formula 3:

b_k is the frequency of residue type k in the background model.

Values $W_{k,j}$ can be:

Positive, when $M_{k,j} > b_k$: the probability of residue k at position j in the motif differs from the background and it is higher (more likely to be an important/functional site than random).

Zero or negative, when $b_k >= M_{k,j}$: the probability of residue k at position j in the motif differs from the background and it is lower (more likely to be a random site than a functional one).

Each sequence X is then scored using a 15-residue sliding window across the first 90 amino acids of the query protein. The score is the sum of the weights over the length L of the sliding window.

$$Score(x|w) = \sum_{i=1}^{L} w_{x_i,i}$$

Formula 4. Scoring sequence X as sum over of sliding window length L.

Value for w is taken from PSWM. The higher score among the 76 (90-15+1) returned by this procedure is stored and compared to a threshold.

This threshold will ultimately discriminate between positive and negative predictions: its optimization is performed with a 5-fold cross-validation. At each iteration three out of five cross validation sets where used to compute PSWM, while one was used as validation test (allocated for the threshold optimization) and last one was used for testing before moving to the benchmark dataset, which represents final testing phase for our model. The threshold is calculated with the precision_recall_curve function of sklearn.metrics module: the maximization of the F1 score will return the optimal threshold.

Each one of these thresholds, after being evaluated on the testing set, will be averaged to obtain the global threshold for the benchmarking set analysis.

2.3 SVM model

There are many good classification techniques in the literature including artificial neural networks, k-nearest neighbors classifier, decision trees, Bayesian classifier and Support Vector Machine (SVM) algorithm[9].
From these techniques, SVM is one of the best-known techniques to optimize the expected solution. SVM algorithm is one of supervised machine learning algorithms based on statistical learning theory. The application scope is very broad due to its excellent learning ability[9].

A SVM is a supervised machine learning algorithm aiming to find the optimal separating hyperplane between two or more classes of data. Such hyperplane is found by maximizing the margin, the distance between the hyperplane and the nearest data points, called support vectors.
SVM can be used on both linear separable and non linear separable data. SVM can be successfully used for linear and non-linear classifications with the aid of kernel functions mapping data to a higher dimensional space, such as polynomial or radial basis function (RBF) (Formula 5).

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

Formula 5. The Gaussian kernel (Radial-Basis Function)

SVM can be implemented with hard or soft margins: in the first case, no data point is allowed inside the margins, while the latter case allow some misclassification, thus being more robust with respect to outliers. The flexibility of the soft margin implementation is granted by a variable ξi called slack variable and the hyper-parameter C, that acts as a trade-off between margin maximization and data fitting.

$$w(\alpha) = \sum_{1=1}^{n} \alpha_i - \frac{1}{2} \sum_{j=1}^{l} \sum_{j=1}^{l} \alpha_1 \alpha_j y_i \gamma_j \langle x_i, x_j \rangle$$

Formula 6. The Dual Lagrangian to be maximized in the soft-margin implementation.

The goal is to maximize Dual Lagrangian which is subject to following constrains:

$$0 \leq \alpha_i \leq C \quad , \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$

Formula 7. Constrains of Dual Lagrangian

Implementation of SVM for our dataset looked like this: The data points in this case, query proteins are mapped into a N dimensional space, with N being the number of features chosen for the evaluation.
Base model consisted of encoding the protein-sequence into a 20-dimensional vector corresponding to the composition of the first k residues of the protein sequence. On top of this all-other relevant features can be encoded and added to the matrix.

It is then of crucial importance to assess which features can explain the relationships within the data. While the von Heijne method takes advantage of the residue composition only, SVM can include many more features such as hydrophobicity, charge and alpha helix propensity.

Each one of these properties can help in distinguishing a SP sequence from a non-SP:

for example, more non-polar residues have to be expected in SP sequences than in the background SwissProt distribution, while charged residues are quite uncommon in SP sequences.

In this implementation on top of the base model, hydrophobicity as feature was added which consisted of following three values for each example (three columns):
global average value, maximal value and peak position of the maximal value over the first k N-terminal residues.
The following ProteinAnalysis class of the Bio.SeqUtils.ProtParam module is able to convert a sequence string into a sequence object to be further analyzed with the scale and sliding window desired: 5 residues in the case of hydrophobicity. Kyte-Doolittle hydropathicity scale[11] was used for mapping hydrophobicity values for each amino acid.
All three values for features have been rescaled to a 0-1 range using the Min-Max scaling.

Several hyperparameters must be set in advance to the training procedure. The kernel coefficient gamma (γ) is a hyperparameter, as well as the regularization parameter C: the length k of the sequence is also a hyperparameter.
Grid search method was applied for selecting the optimal hyperparameter values from pre-defined list.

K = 20, 21, 22, 23, 24
C = 1, 2, 4, 8
γ = 0.5, 1, 2, "scale"
The choice of the kernel type is set to RBF. The SVM computation is carried out using the SVC class of the SVM class provided by the sklearn library.

2.4 Performance measures

Performances of the models have been evaluated by using five metrics which are briefly described below:

Accuracy is a metric that computes the ratio of the correct predictions among all the predictions. Accuracy is defined as:

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

Recall is a metric that indicates how many of the actual positive instances have been correctly identified by the classifier. Recall is defined as:

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

F1 score is the harmonic mean between precision and recall: it assumes high values with certain combinations of precision and recall. F1 score is defined as:

$$\mathrm{F}_1 = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} * \mathrm{Recall}}$$

Precision is a metric that indicates how many of the predicted positive instances are actually true positives. Precision is defined as:

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

Matthews Correlation Coefficient (MCC) is a quality measure for classifications, particularly useful in case of imbalanced datasets. MCC gives

much broader picture of classes, because it's not affected by dataset unbalance, MCC is near 0 in case of random cases, it is 1 in case of perfect prediction and -1 in case of completely wrong prediction. MCC is defined as:

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

3 Results

3.1 Cross Validation Results

As previously said, five-fold cross validation for Von Heijne was used to determine the optimal threshold on the validation set and then test it on the testing set.
This was the chosen value 6.25± 0.52, as optimal threshold, it was chosen as average between five optimal thresholds that were set during cross validation. This threshold was used in further analysis on benchmark set.
Other CV results for Von Heijne are represented in table together with SVM.

For SVM, five-fold cross validation served as optimization process for selecting optimal values three hyperparameters (k, gamma, c).
These values were searched through grid search process where all possible combinations of hyperparameters were tested for each CV set. This means that in five training runs, 400 models were created (80 models for each run). These models were evaluated with MCC metric, and the top best in each run were recorded.
Most frequent values of hyperparameters that gave models with best MCC were chosen as optimal ones. In my case that was following values: K=20, Gamma=2, C=2.

These optimal values will be were used to test models on the testing set (in case of base model and base + hydrophobicity model) and will be used in further analysis on the benchmark.
The following table shows results for both Von Heijne and SVM.
SVM 1 displays cross validation results for base model (just composition of sequences), while SVM 2 displays results for base model + additional feature for hydrophobicity (composition of sequences + 3 values for hydrophobicity).

Table 1. CV Results for Von Heijne, SVM 1, SVM 2

Metrics	Von Heijne	SVM 1	SVM 2
Accuracy	0.93±0.007	0.96±0.004	0.96±0.007
Recall	0.69±0.057	0.79±0.031	0.82±0.087
Precision	0.69±0.070	0.82±0.025	0.84±0.051
F1	0.68±0.009	0.81±0.023	0.82±0.042
MCC	0.65±0.013	0.78±0.025	0.81±0.044

As it can bee seen from Table (1), SVM 1 outperforms Von Heijne which is expected because SVM is one of newer machine learning models while Von Heijne is statistical model that gave decent results when machine learning and neural networks were not so developed.
Also, it can be noted that if we compare SVM 1 to SVM 2, the additional feature does help and gives better overall results among 4 metrics.

3.2 Benchmarking Results

Benchmark results will show the true performance of our model, since it is set that has never been seen before it will give us great insight of how well our model generalizes.

Which is a very important topic especially to the machine learning techniques which contains really powerful algorithms but can tend to overfit if not careful.

For benchmark dataset testing, models were retrained now using all five cross validation sets, for Von Heijne the optimal threshold was used, while for SVM optimal hyperparameters found through grid search were used. The metric results of benchmark analysis were shown in the table (2).

Table 2. Benchmark Results for Von Heijne, SVM 2

Metrics	Von Heijne	SVM 2
Accuracy	0.94	0.97
Recall	0.80	0.82
Precision	0.73	0.89
F1	0.76	0.85
MCC	0.73	0.84

As it can be seen, SVM outperforms Von Heijne for classification of benchmark set. Of course, this is expected as it was shown that SVM did better also on cross validation testing. The interesting thing is that results on benchmark are higher than the ones on the cross validation.

The reason for this could be because the model is retrained on all 5 sets not just three as it was case in cross validation analysis, so it has more examples from which can get clearer picture.

Also benchmark set had 2088 examples, so in this case benchmark set was around 1/4 of the size of training set, while during cross validation, testing sets were around 1/3 of the training set. This could be an additional reason.

Table 3. Confusion Matrix of the Benchmark results, Von Heijne

	Predicted Negative	Predicted Positive
Actual Negative	1805	64
Actual Positive	44	175

True negative and true positive are correctly classified examples, false negative (44) are examples that are classified as negative but their true class is positive, while false positive (64) are sequences that are classified as positive but are actual negative.

Table 4. Confusion Matrix of the Benchmark results, SVM.

	Predicted Negative	Predicted Positive
Actual Negative	1847	22
Actual Positive	38	181

For analysis of FN and FP two that were missclassifed in both models were checked in Uniport.

False positive evaluation is same for von Heijne and SVM methods. Some non-SP sequences still can have a hydrophobic core at the N-terminus, these are often transit and transmembrane proteins like the two examples that were checked (Uniprot codes: Q8L7E5, Q8N8R5).

False negatives can arise for different reasons, in in the von Heijne method, FN predictions could be caused by a different composition at the cleavage site context with respect to the one stored in the PSWM during the training procedure.

In the SVM method, FN predictions could be caused by a different composition in the first k residues.

Examples for FN that were checked:

O43866 and Q8WUJ3 from Uniprot, for both proteins, their SP were examined.

What I could deduce is that they are not having distinct characteristics that we are expecting, for first example, it should be noted that its SP is 19 residues long and is not following the usual composition for the cleavage site that is our basis in Von Heijne, while second one is 30 residues long and from its composition it would be hard to get

standard composition that we are looking for (higher non polar, lower charged amino acids).

Therefore, we can see clear example of problems with FN predictions that are mentioned above.
It should be noted that we had k in range from 20-24 residues which can also limit our ability to recognize SP with lower or higher number of residues.

4 Conclusion

Aim of project was comparing two different approaches for predicting SP. As results shown SVM outperformed Von Heijne in both cross validation and benchmark analysis, which was expected given that Von Heijne is one of first models for successful prediction. Both methods had FN and FP cases.
So even SVM with some additional feature was not enough for distinguish signal peptides from transit and transmembrane proteins in case of false positive. In case of false negative, the signal peptides varied in length and composition/cleavage site gave a lot of trouble to both models.
To successfully deal with FP and FN more complex architectures and models are required, because even though both models perform well and give good results, to achieve higher performance we need more powerful model that can dissect the nonlinearity in these specific cases for FN and FP.

References

1. Henrik Nielsen, Sùren Brunak and Gunnar von Heijne (1999), Machine learning approaches for the prediction of signal peptides and other protein sorting signals. Protein Engineering vol.12 no.1 pp.3-9.

2. Gunnar von Heijne (1990), The Signal Peptide. J. Membrane Biol. 115, 195-201

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

4. Savojardo C. (et.al,2018) DeepSig: deep learning im proves signal peptide detection in proteins. Bioinformatics. 2018 May 15;34(10):1690-1696.

5. The UniProt Consortium (published online in 2014), UniProt: a hub for protein information, D204–D212 Nucleic Acids Research, 2015, Vol. 43, Database issue.

6. Steinegger, M., Söding, J. , (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol 35, 1026–1028.

7. SwissProt composition: https://web.expasy.org/docs/relnotes/relstat.html

8. Gavin E. Crooks (et.al, 2004), WebLogo: A Sequence Logo Generator. Genome Res. 2004. 14: 1188-1190

9. Mustafa Abdullah, D., & Mohsin Abdulazeez, A. . (2021). Machine Learning Applications based on SVM Classification A Review. *Qubahan Academic Journal*, *1*(2), 81–90.

10. Murty, M.N., Raghava, R. (2016). Kernel-Based SVM. In: Support Vector Machines and Perceptrons. SpringerBriefs in Computer Science. Springer, Cham.

11. Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol. 1982 May 5;157(1):105-32.