Sequence analysis

Signal Peptide prediction

Alberti Marta

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. Several computational methods have been developed to overcome the impracticality of comprehensive experimental signal peptide detection. Here, we compare the vonHeijne method, that relies on weight matrices, with a Support Vector Machine model, which is a supervised machine learning algorithm.

Results: The vonHeijne model scored a benchmark MCC=0.582, while the SVM model achieved a MCC=0.612 and generally performs better according to all the metrics adopted (accuracy, precision, recall, F1-score, MCC). Although the vonHeijne model has been outdated by machine learning approaches, the SVM model proposed is well below other ML models, such as SignalP 5.0, that obtained a MCC in the 0.9 range when tested on the same dataset.

1 Introduction

Protein translocation is critical to ensure the correct interaction partners and chemical environment for the successful fulfillment of the protein function. One of the most common sorting signals are signal peptides (SPs): short sequences of 16 to 30 residues, located at the N-terminus, that target newly synthesized proteins to the secretory pathway in both prokaryotes and eukaryotes (von Heijne, 1990). SPs are transient, as they are cleaved away by a signal peptidase upon translocation, and present three distinct regions: a positively-charged N region (n-region), a hydrophobic core (h-region) and a polar uncharged C region (c-region). The end of the SP is marked by the cleavage site (Martoglio and Dobberstein, 1998). The identification of SPs is a key step to shed light on protein localization and function.

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).

Traditional experimental methods for SP recognition are expensive and time-consuming, hence several computational methods have been developed to overcome the impracticality of comprehensive experimental SP detection. Early computational methods for SP recognition relied on weight matrices (von Heijne, 1983; von Heijne, 1986) or simple feature extraction (McGeoch, 1985). The turning point was in the late '90s, when algorithms based on artificial neural networks (ANNs) were proposed and achieved unprecedented results (Nielsen et al., 1997). Machine learning-based methods such as SignalP 4.0 (Petersen *et al.*, 2011) and SPEPlip (Fariselli *et al.*, 2003) are purely ANN-based; while others combine NNs and hidden Markov models (HMMs) to improve predictions, according to the well-known SP structure (Bendtsen *et al.*, 2004). Recently, the use of shallow NNs has decreased in favor of deeper network architectures such as SignalP 5.0 (Almagro Armenteros *et al.*, 2019), SignalP 6.0 (Teufel *et al.*, 2022) and DeepSig (Savojardo *et al.*, 2018). Deep learning has rapidly become the gold standard in the field of SP recognition. A less popular, although interesting, SP detection approach is the use of Support Vector Machine (SVM) with string kernels. The string kernel is a function to evaluate the similarity between sequences on the basis of their k-mers composition (see 2.5 Support Vector Machine).

Indeed, computational SP prediction has gained high relevance in cell biology research, however it is not free of challenges. The main issue is the discrimination between true SP sequences and other hydrophobic regions, specifically N-terminal transmembrane helices.

The purpose of this paper is to compare the SP detection performance of the 1986 vonHeijne method (von Heijne, 1986) with a machine learning approach based on SVM (Cai *et al.*, 2003). The vonHeijne method scored a benchmark MCC=0.582, while the SVM method, tested on the same benchmark dataset (see 2.3 Benchmark dataset) achieved a MCC=0.612. Although neither method produced outstanding results, the SVM method is to be preferred, as it outperforms the vonHeijne method according to all the metrics adopted (MCC, accuracy, precision, recall, F1. See Table 2).

2 Materials and methods

The data used derive from the SignalP 5.0 dataset (Almagro Armenteros *et al.*, 2019). SP sequences were collected from UniProt Knowledgebase release 2018_04 (UniProt Consortium 2018): the dataset included only reviewed entries (belonging to UniProtKB/SwissProt) and SPs with experimental evidence (ECO: 0000269) for the cleavage site, hypothetical proteins and sequences shorter than 30 residues were discarded.

2.1 Training dataset

The training dataset consists of 1723 eukaryotic sequences, randomly selected from the SignalP 5.0 training dataset, and includes 258 positive examples (sequences with N-terminal secretory SPs) and 1465 negative examples (proteins with a subcellu-

lar location annotated as cytosolic, nuclear, mitochondrial, plastid, and/or peroxisomal in Eukarya and not belonging to the secretory pathway with experimental evidence). All sequences were reduced to the first 50 N-terminal residues.
We conducted a data exploratory analysis on the training positive examples and observed that most SPs are in the length range of 20-25 residues (Fig. 1), this distribution is comparable to the benchmark SP length distribution (see 2.3 Benchmark dataset). The main difference is the abundance of shorter SP sequences in the training set with respect to the benchmark set, however the benchmark set contains SPs below 15 residues that are absent in the training set. Nonetheless, both methods that we will implement will exploit the conservation of the average length of SPs in the training and benchmark sets.

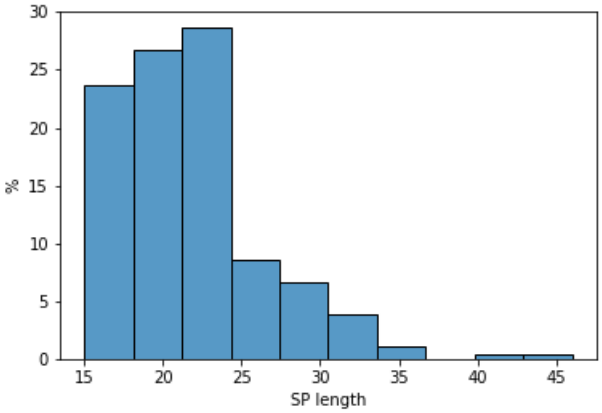

Fig. 1. Histogram distribution of the SP lengths in the training set. Histogram plot showing the length distribution of the training SP lengths. Most SP sequences are in the 20-25 residues length range. The minimum SP length is 15 residues.

We compared the residue composition of the training and benchmark SP sequences with a background composition represented by the residues frequencies in the complete SwissProt dataset (release 2022_04) (Fig. 2). The training and benchmark datasets are very similar and exhibit very noticeable differences with respect to the background SwissProt composition. As expected from the hydrophobic core that characterizes the SP, the training and benchmark SP sequences show an abundance of apolar residues, in particular leucine (L) and alanine (A). The peculiar residue composition and expected SP length will be used to derived a feature encoding for the SVM method to detect the presence of a SP.

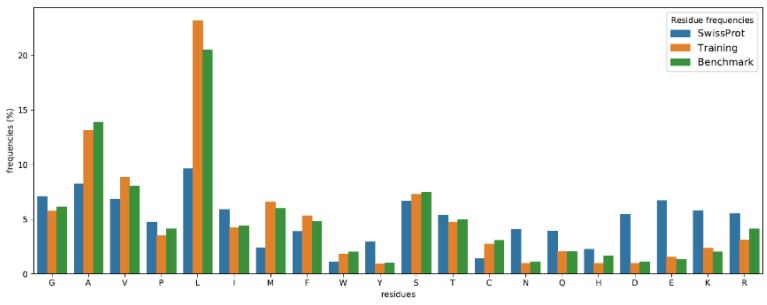

Fig. 2. Signal peptides residue composition comparison between training, benchmark and SwissProt datasets. Training and benchmark SP residue composition are very similar with an abundance of apolar residues, especially leucine and alanine.

We also plotted the [-13,+2] region around the cleavage site (cleavage-site context) of SP sequences with a Sequence Logo (Schneider and Stephens, 1990) (Fig. 3). The Logo allows to capture part of the hydrophobic core: we observe that leucine (L) is highly represented and that there is a conserved AXA motif near the cleavage site (position 0). Unfortunately, this conserved motif is too simple and short to be used on its own to identify SP sequences.

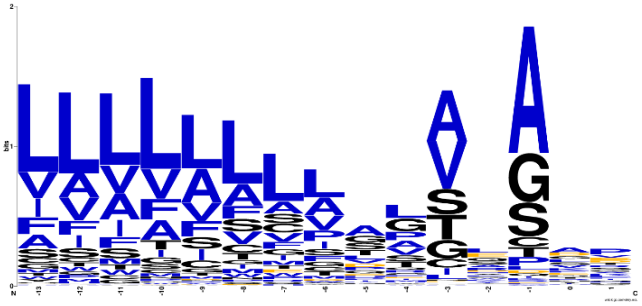

Fig. 3. Sequence Logo of the [-13,+2] region around the cleavage site of training SP sequences. Apolar residues are highly represented, especially leucine (L). The AXA motif near the cleavage site, in position 0, is conserved.

Ultimately, we investigated the taxonomic and kingdom distributions (Fig. 4) and observed that the training dataset is diversified, as it contains SP sequences from several different kingdoms and taxa.

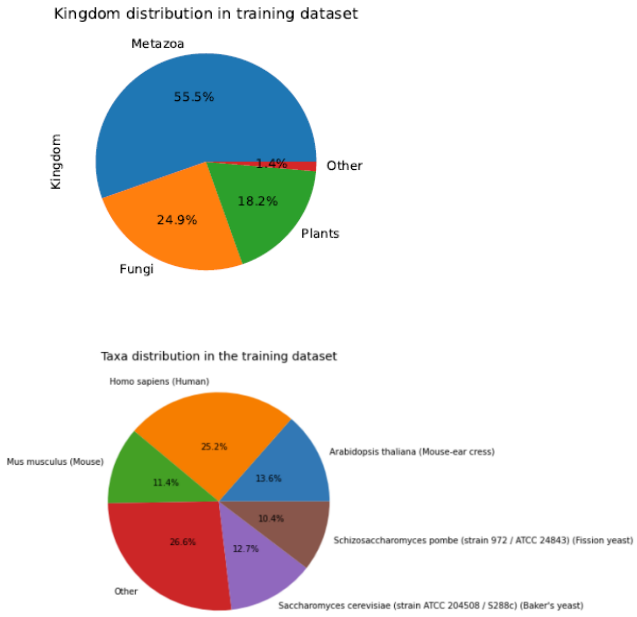

Fig. 4. Kingdom and taxa distributions of training set SP sequences. A wide variety of kingdoms are represented, with a prevalence of Metazoa. Taxa distribution is also heterogeneous with a prevalence of *Homo sapiens*.

2.2 Cross-validation

In order to carry out a cross-validation (CV) procedure, the non-redundant training data was randomly split into 5 folds (from 0 to 4). Each fold contained 345 sequences, except fold 4 which contained 343, both with and without SPs. In the implementation of the vonHeijne method, the 5-fold CV strategy allowed us to pick the optimal sequence score threshold to be used to classify the benchmark sequences as having or not the SP. In the SVM method, CV determined the best combination of hyperparameters to train the model.

2.3 Benchmark dataset

As benchmark dataset we used exactly the SignalP 5.0 benchmark set, composed of 209 positive examples and 7247 negative examples (Almagro Armenteros *et al.*, 2019).
We performed a data exploratory analysis also on the benchmark dataset by plotting the SP lengths distribution (Fig, 5). Most SP sequences are between 20-25 residues long, similarly to the training set. However, in comparison to the training set, in the benchmark set there is a lower percentage of short sequences; also, sequences with a SP shorter than 15 residues are present. All things considered, the benchmark a training SP lengths distribution are comparable and the average length is the same, around 20-25 residues.

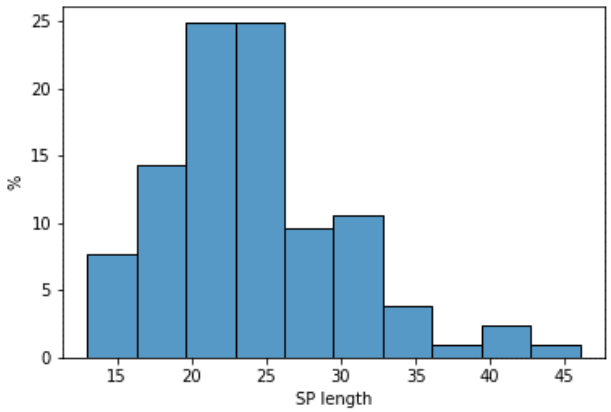

Fig. 5. Histogram distribution of the SP lengths in the benchmark set. Histogram plot showing the length distribution of the benchmark SP lengths. Most SP sequences are in the 20-25 residues length range. With respect to the training set (see 2.1 Training dataset), there are fewer shorter sequences and the minimum length is below 15 residues.

The residue composition of the SP benchmark sequences (Fig. 2) is similar to the SP training sequences, with an abundance of apolar residues such as alanine and leucine.
The benchmark sequence Logo (Fig. 6) describes the abundance of apolar residues, in particular leucine, in the [-13,+2] region around the cleavage site and the AXA conserved motif. From the Logo analysis of the SPs hydrophobic core we determine that the training and benchmark datasets contain sequences sharing the features characterizing the SP: thus making it meaningful to train the vonHeijne and SVM models on the training set and evaluating their performances on the benchmark set.
The taxonomic and kingdom distributions (Suppl. Fig. 1) are comparable to the training set, notably the benchmark dataset is more variegated in terms of represented taxa.

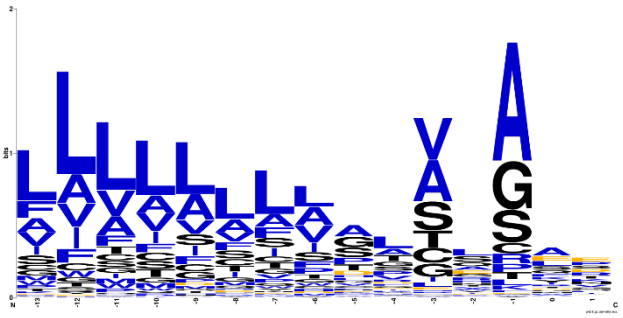

Fig. 6. Sequence Logo of the [-13,+2] region around the cleavage site of benchmark SP sequences. Apolar residues are highly represented, especially leucine (L). The AXA motif near the cleavage site, in position 0, is conserved.

2.4 vonHeijne method

The 1986 vonHeijne method (von Heijne, 1986) for SP detection relies on the modelling of the region around the cleavage site, down to the hydrophobic core (h-region). By characterizing the cleavage site composition and comparing it with a background distribution (e.g. complete SwissProt dataset) we derive a Position-Specific Weigh Matrix (PSWM). A PSWM is used to represent patterns or motifs in biological sequences: it is composed of as many rows as the number of different characters and as many columns as the length of the motif. In our vonHeijne method implementation we care to represent the [-13,+2] region around the cleavage site (15 residues long), thus we will obtain a PSWM with 20 rows, one for each residue in the protein sequences, and 15 columns.
To compute the PSWM, we first extracted the [-13,+2] region around the cleavage site from the training positive sequences. We then computed the Position-Specific Probability Matrix (PSPM) by calculating the frequency of each residue type at each position with the formula:

Equation 1.

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

where, for a set of N sequences: $s_{i,j}$ 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).

A problem that can occur when dealing with finite-size datasets, like in this case, is that in some positions of the sequences not all the residues are observed: thus we would obtain some counts equal to 0, hence the impossibility to compute the log-odds. To avoid zero probabilities in the PSPM we used pseudocounts: we introduced the assumption that each residue in observed at least once in each position by initializing the PSPM with all 1. Using pseudocounts is equivalent to adding 20 artificial sequences to the dataset, hence the N+20 at the denominator in Equation 1.
From the PSPM we computed the PSWM as the logarithm of the PSPM over the SwissProt background frequency:

Equation 2.

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

where b_k is the frequency of residue type k in the background model. The values of $W_{k,j}$ is positive when $M_{k,j} > b_k$, so the probability of residue k in position j in the motif is higher than in the background, so it is likely to be a functional site rather than a random one. When $M_{k,j} \leq b_k$ then $W_{k,j}$ is negative or zero, so the probability of residue k in position j in the motif is lower than in the background, so it is likely to be a random site than an important one.
Once computed the PSWM, it can be used to score motifs in new sequences. Given a sequence X of length L we can calculate the score (log-likelihood) as:

Equation 3.

$$Score_{(X|W)} = \sum_{i=1}^{L} Wx_i, i$$

The values of the scores inform us on the regions of a protein sequence where the likelihood of occurrence of the motif represented by the weight matrix W is highest: we adopted this strategy to score the presence of SPs in the benchmark set. Firstly we computed the PSWM based on the cleavage-site context (region [-13,+2]) of the all the training positive sequences. Then we used the PSWM to score the presence of SPs in the benchmark sequences, using a sliding window of 15 residues over the first 50 N-terminal residues. For each scored sequence we considered the single maximum score, corresponding to the window that most likely represents a SP.
To classify the sequences as having (positive) or not (negative) a SP we needed a threshold on the numerical value of the score, such that all the sequences with score above the threshold are predicted as positive and all those with score below the threshold are predicted as negative. To define such threshold we used a 5-fold CV procedure. We performed 5 independent CV runs: each time we used 4 folds to compute the PSWM and identified the best threshold using a precision-recall curve on the training data. We then classified the sequences in the hold-out set according to the chosen threshold (see 3 Results and Table 1). Lastly, we averaged the thresholds obtained in the CV runs (Suppl. Table 1) to determine the optimal threshold for the benchmark sequences classification (chosen threshold: 8.206).

2.5 Support Vector Machine

A Support Vector Machine (SVM) is a classifier that performs linear separation, in the original or in the feature space, according to the optimality criterion of maximizing the margin (Noble, 2006). The maximum-margin separation can be formulated as a quadratic optimization problem and solved with quadratic programming. The goal is to identify the training points that define the maximum margin separating hyperplane, the so called support vectors. If the data is not perfectly linearly separable we can adopt the soft margin SVM formulation to allow some misclassification errors, this approach involves the additional variable C, that controls the trade-off between margin maximization and fitting the training data (i.e. minimize misclassification errors). In some cases the data is hard non-linearly separable and soft SVM does not work: therefore, to perform the classification, we need to apply a transformation to the original data that maps it into a higher-dimensional feature space where the points become linearly separable. This data mapping is done with a kernel function.

To address the SP detection problem with SVM we adopted a non-linear SVM with Radial Basis Function (RBF) kernel; also, the sequences required to be encoded into a numerical form in order to be presented to the model.
To encode the sequences it was critical to select features that are discriminative of SPs: from the exploratory analysis of the datasets, we know that most SPs are in the range length of 20-25 residues and they also have a distinctive residue composition. Hence, we decided to encode each sequence as a 20-dimensional vector corresponding to the composition of the first k N-terminal residues. The k value, so the length of the N-terminal region to encode, is a hyperparameter that needed to be optimized, as well as the C and the RBF γ parameters. The k, C, γ parameters optimization was done with CV and a grid search: firstly we defined a set of values for each parameter, we then tested each possible combination of values with a complete CV run and selected the combination achieving the highest performance, according to the Matthews Correlation Coefficient (MCC) (Suppl. Table 2).

Equation 4.

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

With the grid search procedure we tested the following values: {20, 22, 24} for k, {1, 2, 4} for C and {0.5, 1, 'scale'} for γ. The best performing combination resulted to be k=20, C=2, γ='scale', scoring a MCC=0.840.

2.6 Performance metrics

To evaluate the models performance in predicting the presence of SPs in the benchmark sequences, we compared the models predictions with the real classes: class '0' for sequences without SP and class '1' for sequences endowed with a SP and computed the confusion matrices (CMs) (Fig. 7). We then used the CMs to calculate 5 scoring metrics: accuracy, precision, recall, MCC and F1-score.

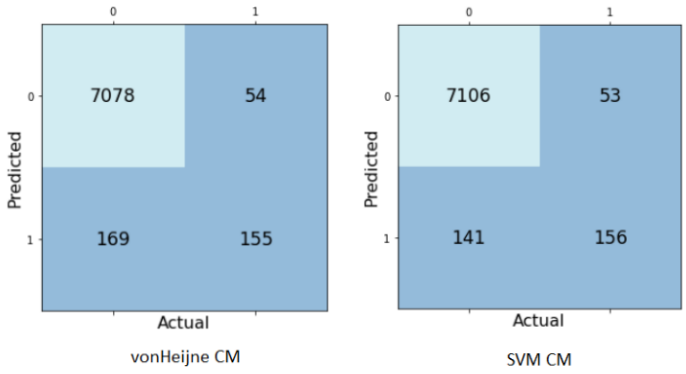

Fig. 7. Benchmark confusion matrices for vonHeijne and SVM models.

Accuracy (Eq. 5) is the ratio of correctly classified examples over the total number of examples.

Equation 5.

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

Accuracy is a very intuitive performance metrics, however it is sensitive to class imbalance, resulting in a misleading interpretation of the results. Our benchmark dataset is indeed unbalanced, as there are only 209 positive examples and 7247 negative examples (see 2.3 Benchmark dataset), hence the use of multiple metrics.
Precision (Eq. 6) is the ratio of correctly classified positive examples over the total predicted positive examples. High precision relates to a low False Positive Rate (FPR).

Equation 6.

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

Recall (Eq. 7) is the ratio of correctly predicted positive examples over the total number of real positive examples. High recall relates to a low False Negative Rate (FNR).

Equation 7.

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

F1-score (Eq. 8) is the weighted average of precision and recall, so it takes into account both the false positives and the false negatives, therefore a classifier will obtain a high F1-score only if both precision and recall are high.

Equation 8.

$$F1\ Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

The MCC (Eq. 4) calculates the correlation between actual classes and predicted classes: the higher the correlation between true and predicted values, the better the predictions are. MCC ranges from -1 to +1: where +1 and -1 respectively indicate perfect agreement and disagreement between predictions and actuals. A MCC=0 denotes that the predictions are random with respect to actuals, so it is a random guessing classifier.
The last metric to introduce is the False Positive Rate (FPR) (eq. 9), that we computed to analyse the false positive cases (see 3.2 False positive analysis). The FPR is defined as the total number of negative examples incorrectly identified as positive cases divided by the total number of negative cases.

Equation 9.

$$FPR = \frac{FP}{FP + TN}$$

3 Results

We adopted a 5-fold CV procedure to optimize the two models. (see 2.2 Cross-validation). The optimization of the vonHeijne method allowed to select the optimal sequence score threshold to classify the benchmark sequences; while for the SVM method, CV determined the best combination of hyperparameters k, C, γ. For each model we performed a 5-fold CV run and computed accuracy, precision, recall, F1-score and MCC, we then averaged the performance scores that each CV run obtained (Table 1).

Table 1. Comparison between vonHeijne and SVM CV scores.

	vonHeijne	SVM
Accuracy	0.945 ± 0.004	0.959 ± 0.006
Precision	0.843 ± 0.031	0.877 ± 0.027
Recall	0.791 ± 0.016	0.852 ± 0.017
F1-score	0.814 ± 0.009	0.863 ± 0.017
MCC	0.784 ± 0.011	0.840 ± 0.020

For the vonHeijne model: each CV run had its own specific threshold used to predict the hold-out set and we averaged the performances of all the 5-fold CV runs on the hold-out. For the SVM model: once we selected the best C, k, γ combination (20, 2, 'scale' respectively) we ran a full 5-fold CV and here reported the average performance.

The CV procedure determined that the optimal sequence score threshold for the vonHeijne model was 8.206, while the optimal combination of parameters for SVM was k=20, C=2, γ='scale'. We used these parameters to make the two models predict the same benchmark sequences and obtained the following results:

Table 2. Comparison between vonHeijne and SVM benchmark scores.

	vonHeijne	SVM
Accuracy	0.970	0.974
Precision	0.479	0.525
Recall	0.742	0.746
F1-score	0.582	0.617
MCC	0.582	0.614

Benchmark scores for vonHeijne (classification threshold = 8.206) and SVM (k=20, C=2, γ='scale').

The SVM model outperforms the vonHeijne model on all the metrics considered, however it is not on par with SignalP 5.0 (Almagro Armenteros *et al.*, 2019), comparison made because the benchmark set is exactly the same, which predicts the presence of SPs in Archea, Gram-negative bacteria, Gram-positive bacteria and Eukarya, and scored an overall MCC between 0.890 (Gram-positive bacteria) and 0.966 (Eukarya).
We then conducted a further analysis on the FP and FN sequences to understand the reason for their misclassification.

3.1 False negative analysis

False negative (FN) sequences are those endowed with a SP but are not recognized as such by the model, thus are classified as negative. The reason for these errors may be related to the assumptions made to implement the model and when some sequences do not meet such assumptions the misclassification occurs.

3.1.1 FN analysis for vonHeijne model

For the vonHeijne model we assumed that SPs can be recognized based on the cleavage-site context, so we compared the training SP sequences Logo (Fig. 3) with the Logo of the TP (Fig. 8 above) and FN sequences (Fig. 8 below). As we can observe, the TP sequences Logo is very similar to the training Logo and indeed they were correctly labeled. Conversely, the FN sequences

Logo is very different: there is no single dominating residue in the positions where the training and TP sequences show a high conservation of leucine (L) and the motif AXA near the cleavage site is poorly conserved.
From this analysis we can conclude that the reason for the FN errors is that some SP sequences in the benchmark set differ from the SP residue composition of the training sequences, so the model is unable to detect the SP presence and results in misclassification.

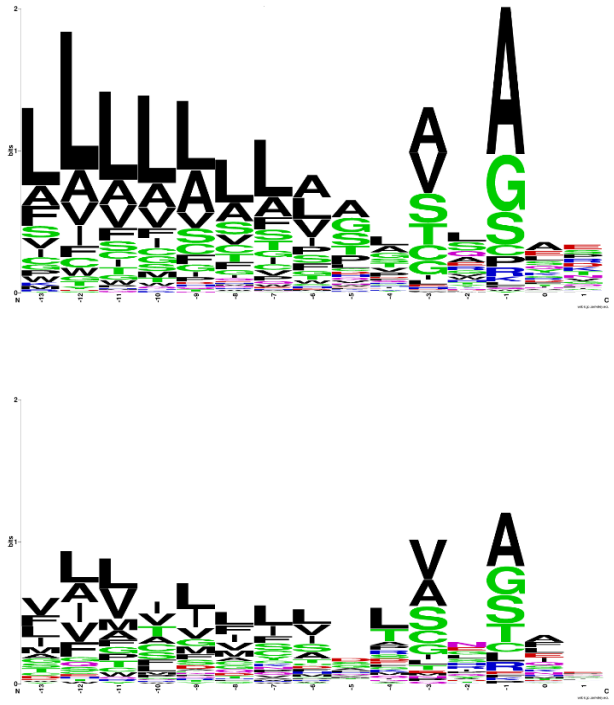

Fig. 8. Sequence Logo of the [-13,+2] region around the cleavage site of vonHeijne TP sequences (above) and vonHeijne FN sequences (below). Sequence Logo of the vonHeijne TP sequences is very similar to the training Logo (Fig. 3) and the motif AXA near the cleavage site is conserved. Sequence Logo of the vonHeijne FN sequences is very different from the training Logo: no single dominating residue and the motif AXA near the cleavage site is poorly conserved.

3.1.2 FN analysis for SVM model

For the SVM model we assumed that SPs can be recognized based on the residue composition of the first *k* N-terminal residues. Therefore, to investigate the FN results, we decided to compare the SP length distribution and residue composition of the SVM TP, FN and training sequences.
The SP length distribution of the TP sequences (Fig. 9 above) is very similar to the training SPs length distribution (Fig. 1): for these sequences the assumption that most SPs are in the length range of 20-25 residues holds and indeed they were correctly classified. Contrarywise, the SP distribution of the FN sequences (Fig. 9 below) is rather different: most sequences are between 30 and 35 residues long and the distribution is more uniform.
This comparison shows that for the FN sequences, modelling the composition of the first k=20 residues causes the inclusion of noise for sequences with SPs shorter than 20 and to miss important parts for sequences with SPs longer than 20 residues.

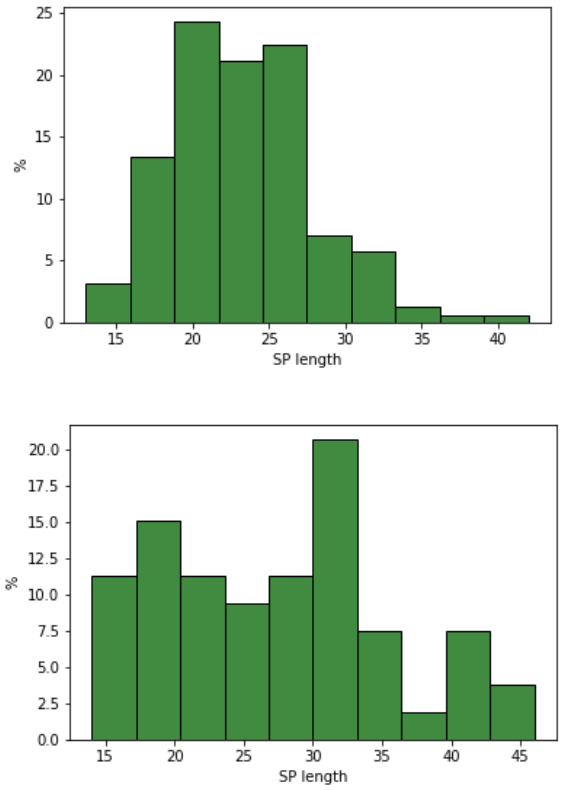

Fig. 9. Histogram distribution of the SP lengths in SVM TP sequences (above) and SVM FN sequences (below). Most SPs in TPs are in the 20-25 residues length range. Mean length = 23.2 residues. Median length = 22.5. Very similar to the training SPs length distribution (Fig. 1). Most SPs in FNs are in the 30-35 residues length range. Mean length = 27.0 residues. Median length = 27.0. Uniform distribution, different from the training SPs length distribution.

Concerning the composition of the first 20 residues, we note that TP sequences are very similar to the training sequences and once again FNs differ (Fig. 10). FN sequences show an abundance of arginine (R) and a scarcity of leucine (L) with respect to the training positive sequences. It is likely that the difference in the SP composition and length distribution caused the misclassification of FN sequences.

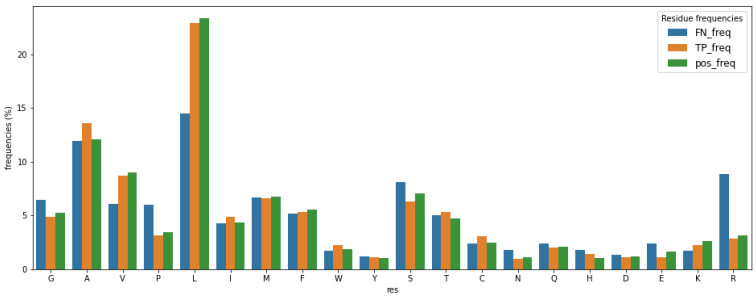

Fig. 10. Residue composition of the first 20 (optimized k) residues in SVM false negative (FN_freq), true positive (TP_freq) and positive training sequences (pos_freq). TP and training sequences residue compositions are very similar. FN residue composition differs as there is an abundance of arginine and scarcity of leucine.

3.2 False positive analysis

The other type of classification mistake that our models can make are false positives (FPs), these errors occur when a sequence without a SP is labeled as positive. To investigate the FP errors we computed the total FPR (Eq. 9) for the two models, which was 2.3% for the vonHeijne model and 1.9% for the SVM model.

As stated before (see 1 Introduction) the main issue in the computational SP prediction is the discrimination between true SP sequences and other hydrophobic regions, specifically N-terminal transmembrane helices (TM helix), but also different types of sorting signals such as transit peptides (TPs), that target the proteins towards the cell organelles. In light of this we used the UniProt ID Mapping tool to display the annotations related to TM helices and TPs for all the negative benchmark sequences. We then computed the FPR specific for TM helices and for the various kinds of TPs (Table 3).

Table 3. FPR comparison between vonHeijne and SVM models.

FPR (%)	vonHeijne	SVM
Total	2.3	1.9
TM helix	28.4	27.7
Total TP	3.6	3.5
Mit TP	3.4	4.7
Chl TP	3.9	2.2
Perox TP	0	0

FPR comparison between vonHeijne and SVM. Transit peptide (TP) to direct the protein to mitochondria (mit TP), chloroplast (Chl TP) or peroxisome (Perox TP). TM helix = transmembrane helix.

We noted that the FPR for TM helices is 12 times higher than the total FPR for the vonHeijne model and 14 times higher for the SVM model. This clearly indicates that both methods have a propensity to label a SP-free sequence as positive when a TM helix in the first 50 N-terminal residues is present. Since TM helices are the main source of misclassification we should include in our models a specific strategy to address this issue.

4 Conclusion

The goal of this work was the comparison of the vonHeijne and SVM methods for SP detection. To do so, we trained and tested each model on the same training and testing sets (see 2.1 Training dataset and 2.3 Benchmark dataset). We optimized each model via a 5-fold cross-validation procedure (see 2.2 Cross-validation) in order to select the optimal sequence score threshold for benchmark classification for the vonHeijne model (see 2.4 vonHeijne method) and to determine the best combinations of k, C, γ parameters for the SVM model (see 2.5 Support Vector Machine). The models comparison was conducted by exploiting 5 performance metrics: accuracy, precision, recall, F1-score and MCC (see 2.6 Performance metrics). According to all these metrics the SVM model performs better than the vonHeijne one, however it is outplayed by SignalP 5.0. The false negative analysis showed that the vonHeijne model misclassified those benchmark sequences that presented a more homogenous SP residue composition with respect to the training SPs (see 3.1.1. False negative analysis for vonHeijne model). The SVM model also struggled to correctly classify the benchmark sequences whose SP did not present the canonical SP features (abundance of leucine and scarcity of arginine) and, since it considered only the 20 N-terminal residues, another cause of misclassification was a shorter or longer SP (see 3.1.2 FN analysis for SVM model). For both the vonHeijne and SVM models the main reason for false positive errors is the presence of transmembrane helices in the first 50 N-terminal residues. In conclusion the vonHeijne model has been outdated by ML approaches, however the proposed SVM model is particularly sensitive to the presence of transmembrane helices, thus it is not on par with more sophisticated models such as SignalP 5.0.

References

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

Cai,Y.-D. *et al.* (2003) Support vector machines for prediction of protein signal sequences and their cleavage sites. *Peptides*, **24**, 159–161.

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.

Dyrløv Bendtsen,J. *et al.* (2004) Improved Prediction of Signal Peptides: SignalP 3.0. *Journal of Molecular Biology*, **340**, 783–795.

Fariselli,P. *et al.* (2003) SPEPlip: the detection of signal peptide and lipoprotein cleavage sites. *Bioinformatics (Oxford, England)*, **19**, 2498–2499.

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

Martoglio,B. and Dobberstein,B. (1998) Signal sequences: more than just greasy peptides. *Trends in Cell Biology*, **8**, 410–415.

McGeoch,D.J. (1985) On the predictive recognition of signal peptide sequences. *Virus Research*, **3**, 271–286.

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

Nielsen,H. *et al.* (1997) Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering Design and Selection*, **10**, 1–6.

Noble,W.S. (2006) What is a support vector machine? *Nature Biotechnology*, **24**, 1565–1567.

Petersen,T.N. *et al.* (2011) SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nature Methods*, **8**, 785–786.

Savojardo,C. *et al.* (2017) DeepSig: deep learning improves signal peptide detection in proteins. *Bioinformatics*, **34**, 1690–1696.

Schneider,T.D. and Stephens,R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, **18**, 6097–6100.

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 Se-

lective Inhibitor of Co-Translational Protein Translocation. *PLoS Biology*, **12**, e1002011.
von Heijne,G. (1986) A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research*, **14**, 4683–4690.
von Heijne,G. (1983) Patterns of amino acids near signal-sequence cleavage sites. *European Journal of Biochemistry*, **133**, 17–21.
von Heijne,G. (1990) The signal peptide. *The Journal of Membrane Biology*, **115**, 195–201.
UniProt Consortium. Nucleic Acids Res. 46, 2699 (2018).