Comparison between PSWM and SVM based models for signal peptides detection

Edoardo Bettazzi

International Master in Bioinformatics, University of Bologna

Abstract

Motivation: Amino-terminal signal peptides (SPs) are short regions that guide the targeting of secretory proteins to the correct subcellular compartments in the cell. Detecting their presence in protein sequences can provide useful information about subcellular localization, a key feature in the process of functional annotation. Since the common structure of SPs has been known for a long time, several methods have been developed, employing very different approaches for the detection of signal peptides. Here, we present a comparison between two common approaches: we developed a position-specific weight matrix (PSWM) model based on the method by (von Heijne, 1986) and a supervised learning model based on Support Vector Machines (SVMs). The models were then trained and evaluated on the SignalP-5.0 datasets by (Almagro Armenteros et al., 2019).
Results: Our results indicate that the SVM-based model outperforms the PSWM-based model in the detection of SPs, with an MCC of 0.612 against an MCC of 0.582.

Code Availability: The implemented pipeline and all the materials used in the paper are available at https://github.com/EdoardoBettazzi/lb2-2022-project-Bettazzi.git
Contact: edoardo.bettazzi@studio.unibo.it
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1. Introduction

Subcellular protein sorting, i.e. the processes through which proteins are routed to their proper final destination within a cell, is a fundamental aspect of cellular life. In many cases, sorting depends on "signals" that are encoded in the primary structure of proteins, enabling the cellular machinery to target and deliver them with outstanding selectivity and specificity. The sorting code is mostly represented by N-terminal stretches of amino acids, 18 to 26 residues long, called signal peptides (SPs) (Hajar, 2018). SPs are found in virtually all organisms across the life domains, with the same role of targeting the newly synthesized proteins to secretory pathways, directing translocation across the plasma membrane in prokaryotes, and across the endoplasmic reticulum membrane in eukaryotes. During or after membrane translocation, the signal peptide is removed by a signal peptidase (SPase) with the cleavage site (CS) having a particular structure in Archaea, Bacteria and Eukarya (Perlman & Halvorson, 1983). Apart from this, the SPs have a general common structure with three defined regions: a positively charged region at the N terminus, followed by an hydrophobic core, and a neutral, polar C-terminal region (Hajar, 2018). Despite the complexity of the task, the structural differences in proximity of the CS introduce the possibility of discriminating proteins and secretory pathways that belongs to different life domains. For the relevant contribution that this can give to the functional annotation of proteins as well as cellular processes, the problem has long captured interest. Since it is fundamentally a problem of natural language processing (NLP), where a CS motif, like a word in a sentence, needs to be recognized in a longer sequence according to its meaning, machine learning (specifically, deep learning) methods are very suitable to tackle it. Taking advantage of the continuous developments in the field of NLP, many models are being published year by year. As an example, a recent advancement was made by (Teufel et al., 2022), whose model SignalP 6.0 has outperformed other 16 models in the prediction of SPs in Archaea, Bacteria and Eukarya. Considering the state of the art, any increase in performance is an outstanding result, and benchmarking studies give a context to this, taking into account the best performing tools in the field. In that sense, it is meaningless to include methods that are not employed any more for the task under study, even though those methods can give a substantial, almost historical, meaning to the novelty and the advancement of new tools. Therefore, here we present a comparison between two different approaches to the signal peptides detection problem. In particular, we were focused on one of the first methods for detecting SPs, designed and developed by (von Heijne, 1986), that is based on position-specific weight matrices, and compared it with a method generally employed in classification tasks, and especially useful in the field of bioinformatics, that is Support Vector Machines (SVMs) (Cortes & Vapnik, 1995).
A Position Specific Weight Matrix (PSWM) is basically a motif descriptor. It represents, for each position of the pattern, the probability of a given character occurrence

(residue, for proteins) weighted against a general background. The weight is computed as a logarithm, resulting in a negative value wherever the residue is less frequent than it is in the background (i.e random occurrence), and a positive value when the residue is more frequent. Summing up weights for a given sequence, results in a score that can be compared with optimized thresholds to discriminate between sequences that contain a relevant signal, and sequences that don't (von Heijne, 1986).
Support Vector Machines (SVMs) are conceptually different. Taking as input a vector of the features, in this case a vector of residue frequencies, an SVM maps the input vector into some high dimensional feature space through some non-linear mapping chosen a priori. In this space a linear decision surface is constructed so that data, sequences, can be separated and classified as presenting a given signal or not (Cortes and Vapnik, 1995).
We trained two models based on the described methods, and evaluated their performance. The datasets used for the study are derived from the SignalP-5.0 datasets by (Almagro Armenteros et al., 2019). Results show a superior performance by the SVM model, with a Matthew correlation coefficient (MCC) of 0.612 and false positives rate (FPR) of 0.0032, while the von-Heijne model performed with an MCC of 0.582 and FPR of 0.0233.

2. Materials and Methods

2.1. Dataset

The datasets used for training and benchmarking were derived from the SignalP-5.0 dataset by (Almagro Armenteros et al., 2019). The authors extracted the sets from the UniProt Knowledgebase release 2018_04. Only reviewed entries (SwissProt) and signal peptides with experimental evidence (ECO:0000269) for the cleavage site were included. Every sequence with length <30 AAs was discarded. The training set we used consists of a randomly selected subset of the original eukaryotic SignalP-5.0 training dataset, counting 1723 sequences divided in 258 positive examples (i.e. sequences endowed with N-terminal secretory signal peptides) and 1465 negative examples (i.e 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). Moreover, the dataset was randomly split into 5 equally-size different subsets (345 sequences per each, except one subset counting 343), in order to perform a 5-fold cross-validation procedure. It is worth mentioning that the original dataset is non-redundant (30% maximum similarity and 40% alignment coverage), therefore the split could be performed randomly without introducing biases in the subsets. The benchmark set is the same dataset used for benchmarking SignalP-5.0 and the other approaches, counting 7456 eukaryotic sequences, divided into 209 positive examples and 7247 negative examples. In both sets, all sequences were shortened to the first 50 N-terminal residues. Statistical analyses of the datasets showed similar results in both datasets. Plotting the length distribution of SP sequences revealed that most had a length in the range of 20 to 25 residues (Fig. 1a and 1b, Supplementary materials); a compositional analysis confirmed an abundance of apolar residues (Leucine and Arginine, Fig 2a and 2b, Supplementary materials), as expected by the hydrophobic composition of the core. Finally, observing the taxonomic distribution, it showed a prevalence of sequences from Metazoa with respect to the other kingdoms, but an heterogeneous distribution when classifying by taxa (Fig 3a and 3b, Supplementary materials). Finally, the Sequence Logo (Schneider & Stephens, 1990; Crooks et al., 2004) in the region [-13,+2] around the cleavage site revealed an abundance of Leucine, as well as the strong conservation of the canonical AxA cleavage motif (Fig 4a and 4b, Supplementary materials).

2.2. von-Heijne method

The von-Heijne method (von Heijne, 1986) consists in the use of PSWM to model the hydrophobic region around the cleavage site, specifically in the region [-13, +2], of a given representative set of SPs sequences. The PSWM trained on the representative set is then used to score new sequences and discriminate them based on a threshold.
In order to build the model, we first extracted the signal peptide sequences (region [-13, +2]) from the training set. Then, we computed a position specific probability matrix, based on the frequency of occurrence of each residue in each position of the set of SPs. In order to avoid zero probabilities in the PSPM, and hence the impossibility of computing the logarithms, pseudocounts of 1 were added during the computation of the PSPM, assuming that every residue was present at least once in the alignment. Formally, the PSPM was computed according to:

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

where:

- $s_{i,j}$ 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)

From the PSPM, the position-specific weight matrix was obtained by computing the logarithm of the probability matrix over a background model (the AA composition computed on the entire SwissProt database), according to:

$$W_{k,j} = log\frac{M_{k,j}}{b_k} \quad (2)$$

where b_k is the frequency of the residue *k* in the background model.

A positive value of $W_{k,j}$ for a residue k in position j indicates that this site is more likely to be associated with a signal peptide, rather than chance (random probability, represented by the background). Negative values indicate the opposite. Once built, a PSWM can be used to compute a score (log-likelihood) of any sequences. Formally, given a sequence X of length L and a PSWM M, the score is computed as:

$$Score_{(X|W)} = \sum_{i=1}^{L} W_{x_i}, i \quad (3)$$

Considering a window of 15 residues (as the selected region around the cleavage site), each sequence (50 residues in length) of the training set was scanned to find the region where the likelihood of the motif

occurrence was maximized. The computed score was associated with the sequence itself. To perform a classification of positives (i.e having the SP) or negatives (i.e not having the SP), the score needs to be compared with threshold that allows for a safe discrimination. Therefore we performed a 5-fold cross-validation procedure to obtain an optimal threshold. The mean value of the thresholds obtained from the cross-validation runs (see Supplementary materials) was selected as optimal, and was used to evaluate the model performance on the benchmark set.

2.3. Support Vector Machines

Support vector machines (SVMs) are a supervised learning method that, upon training on a dataset, allows the separation of data points into two (or more) classes. It performs the classification in linearly separable datasets, by setting a decision boundary, an hyperplane on the feature space, that maximizes the margin, that is the distance, between the data belonging to different classes.
By setting a hyperparameter (C), the margin can allow for misclassification of difficult or noisy training points. Low values of C allows for more flexibility during training, and the margin is called soft; high values of C makes the model more rigid toward misclassification, and the margin is said to be hard. When dealing with non-linearly separable datasets, SVM maps the data points into a high-dimensional space where a linear classification is possible. Since computing explicit transformations to high-dimensional spaces can be challenging, SVM applies kernel functions that map data implicitly to a feature space that depends on the chosen kernel.
To develop our SP detecting model, we used the SVM classifier implemented in the scikit-learn library (F. Pedregosa et al., 2011). Each training sequence was encoded in a 20-dimensional frequency vector corresponding to the first k residues. The length of the encoded sequences (k) was optimized on the basis of the length distribution of the SPs in our training set. To deal with the non-linearity of the task, the radial basis function (RBF) was selected as kernel function, due to its capacity to generalize well to a wide set of problems. Moreover, the RBF kernel requires the optimization of only two hyperparameters: the regularization term C, already introduced, and the kernel coefficient "gamma", that controls the spread of the decision region. Except for these two, every other parameter was set to default. To search for the optimal combination of hyperparameters, a grid search was performed during a 5-fold cross-validation. The following values were considered: k ranging from 19 to 24; C ranging from 1 to 20; gamma, considering the default 'scale' and values ranging from 0.1 to 1 with decimal steps. The optimal combination was selected based on the highest MCC, and used for the performance evaluation on the benchmarking dataset.

2.4. Scoring measures

The above models were evaluated, both in the training and in the benchmarking phases, according to the following measures:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (8)$$

$$FPR = \frac{FP}{FP + TN} \quad (9)$$

The Matthew correlation coefficient (MCC) was used as the main measure for the evaluation of performance as it depends on the four categories of results, without suffering from class imbalances or biases in the dataset (as accuracy and F1), therefore it reports the quality of binary classification in a reliable way (D. Chicco & G. Jurman, 2020).

3. Results

The 5-fold cross-validation procedure, performed to optimize the models, resulted in the following hyperparameters optimal values: mean threshold of 8.206 for the von-Heijne model; and k = 19, C = 16, gamma = 0.8 for the SVM model. All the associated metrics can be found listed in Table 1, below.

Table 1. 5-fold cross-validation results for both models

	von Heijne	**SVM**
MCC	0.784 +/- 0.011	0.858 +/- 0.013
Accuracy	0.945 +/- 0.004	0.964 +/- 0.004
Precision	0.843 +/- 0.036	0.907 +/- 0.028
Recall	0.791 +/- 0.016	0.853 +/- 0.017
F1	0.816 +/- 0.009	0.877 +/- 0.011

After optimization, the two models were tested on the benchmark dataset for the evaluation of performances, where the von-Heijne method performed with an MCC of 0.582, while the SVM scored 0.612. The overall results are listed in Table 2, below.

Table 2. Comparative benchmarking results for both models

	von Heijne	**SVM**
MCC	0.582	0.612
Accuracy	0.970	0.982
Precision	0.478	0.811
Recall	0.742	0.474
F1	0.582	0.598
FPR	0.023	0.003

The SVM model seems to outperform the von-Heijne model, except for the recall metric, which measures the sensitivity of the model to true positives. Considering that the C hyperparameter has a value of 16, that is quite high, the SVM model largely penalizes misclassifications, therefore it is cautious when classifying sequences as positives. This translates into lower sensitivity, but also a very low rate of false positive results. In line with this, it can be noticed that the precision measure is instead high. To further characterize the models, an analysis of the false results was conducted.

4. False positives analysis

We speculated about two possible sources of error: first, the presence of a transmembrane domain in the first 50 residues, given the hydrophobic composition and N-terminal location; second, the presence of transit peptides (TP), less hydrophobic in composition, but still predictable as signal peptides. Therefore, the FP predictions in the benchmark set were isolated, and using the protein IDs and UniProtKB, the FP rates for each of the mentioned categories were computed. Since transit peptides have a slightly different composition depending on the targeted organelle, the rates were computed also for mitochondrial, chloroplastic and peroxisomal TPs. The results can be found in Table 3, below.

Table 3. Results of the false positive rates (FPR) analysis

	von Heijne	**SVM**
False positive rate	0.023	0.003
Transmembrane proteins	0.297	0.115
Transit peptides	0.035	0.001
Mitochondrial TPs	0.033	0.001
Chloroplastic TPs	0.040	0.0
Peroxisomal TPs	0.0	0.0

As supposed, the main source of error is the presence of transmembrane domains at the N-terminus.

5. False negatives analysis

To investigate the reasons behind false negative results, we took into consideration the assumptions of the two methods. The von-Heijne method focuses the training of the model on a selected region (+13,-2) around the cleavage site, therefore the FNs may be due to a different composition of the cleavage-site context. To check for this hypothesis, we computed the Sequence Logo for all FNs in the benchmark set and compared it with the Sequence Logo of the positives of both benchmark and training set (Fig. 5A, Supplementary materials): the comparison showed that the AxA motif was not present in the FNs, where the Alanine is substituted with a Valine; moreover, in the fourth position Leucine is substituted with a Threonine, and overall the trail of Leucines that characterizes the Logo in the datasets seems to be flattened in the false negative results. This confirms that the von-Heijne method suffers from more homogenous composition as they might lack signals relevant to the detection. Despite that, a deeper analysis would be necessary to identify the positions that make the model more vulnerable to misclassifications. The SVM method relies on the composition of a region with a given length, therefore the false negatives may be due to a variation in composition as well as a variation in the length of the misclassified signal peptides. To investigate the hypotheses, the length distribution of the FNs was compared with the length distribution of the positives of the benchmark set, but there was no significant difference (Fig. 5B, Supplementary materials). On the other hand the compositional analysis comparing false negatives, true positives and training positives, revealed slight differences (Fig. 5C, Supplementary materials) that could orient a rigid model (as it is our, due to the high value of the regularization term) toward the classification of positive data points in the negative category.

6. Conclusion

We developed two models based on different approaches to the problem of signal peptides detection: one model was based on the von-Heijne method, that relies on the building of a position-specific weight matrix (PSWM), while the other was based on the SVM method, which relies on supervised machine learning. Both were trained and optimized with a 5-fold cross-validation on a training set derived from the SignalP-5.0 dataset. The models were then tested on the benchmark set from the same dataset. The performances were evaluated and compared on the basis of the Matthew correlation coefficient (MCC) and false positive rates (FPR). Our results indicate that the SVM method outperforms the von-Heijne method with an MCC of 0.612 against an MCC of 0.582. Moreover, the SVM model had an FPR of 0.003, almost a factor of 10 lower than the von-Heijne model FPR (0.023). We investigated the reasons behind the false results of each model, with the presence of transmembrane domains and the variability in the AA composition of the sequences being the shared and most significant sources of error. In conclusion, the Support Vector Machines seem to be more fit for the signal detection task, even though the performance is not exceedingly higher than that of the von-Heinje method. For this reason, it would be interesting to further investigate this comparison, possibly using datasets specifically crafted to fill in the weaknesses of the methods and exploit their potential.

References

Almagro Armenteros,J.J. et al. (2019) SignalP 5.0 improves signal peptide predictions using deep neural networks. *Nature Biotechnology* 2019 37:4, 37, 420–423.

Cortes,C. and Vapnik,V. (1995) Support-Vector Networks. *Mach Learn*, 20.

Chicco, D., Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020).

Crooks,G.E. et al. (2004) WebLogo: a sequence logo generator. *Genome Res*, 14, 1188–1190.

Hajar Owji, Navid Nezafat, Manica Negahdaripour, Ali Hajiebrahimi, Younes Ghasemi, A comprehensive review of signal peptides: Structure, roles, and applications, *European Journal of Cell Biology*, Volume 97, Issue 6, 2018, Pages 422-441, ISSN 0171-9335.

Pedregosa F. et al. (2011) Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. *Journal of Machine Learning Research*, 12, 2825–2830.

Perlman D, Halvorson HO. A putative signal peptidase recognition site and sequence in eukaryotic and prokaryotic signal peptides. *J Mol Biol*. 1983 Jun 25;167(2):391-409.

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

Teufel, F., Almagro Armenteros, J.J., Johansen, A.R. *et al.* SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nat Biotechnol* **40**, 1023–1025 (2022).

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