Rostlab
Problem-Based Learning
Publication Date: 31 January 2020
Final report

Final report

Signal peptide prediction using Sci-kit Learn

Anonymous*, My Supervisor[1] as supervisor

[1]Department of Informatics, I12—Chair of Bioinformatics and Computational Biology, Technical University of Munich (TUM), Boltzmannstrasse 3, 85748 Garching/Munich, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: In recent years research has been facing a growing demand for biotherapeutics, in particular for recombinant proteins. Production of the latter however remains a challenge, as proteolytic processing and incorrect protein folding are among many issues which lead to a loss in final product quantity. A possible way of improving this common situation is provided through increasing the protein translocation rate and secretion efficiency by tuning the respective signal peptide. Not only are these N-terminal sequences involved in the production of medication, they also serve as targets of drugs. A promising example are specifically designed therapies which tackle the signal peptides of proteins originating from malaria parasites. The difference between their signal peptides and the human equivalent is the necessary condition enabling this emerging method of battling the malaria disease. Hence, computational prediction of signal peptides in protein sequences has become of great interest and numerous algorithms have been developed.

Results: Here, we want to give an overview over the complex subject of signal peptides while presenting an introduction to using FASTA files for classification tasks and building a simple, yet effective, machine learning model. For this we used Scikit-learn to set up several machine learning instances as well as for calculating the performance metrics and statistical curves. In addition Imbalanced-learn supplied necessary tools for imbalance correction. In order to keep the redundancy found in the datasets provided by SignalP 5.0 as low as possible, we implemented cross-validation manually to preserve the partitions already given by the datasets. Its nested form ensures a maximized use of both datasets. The beneficial effect of the manual implementation of cross-validation becomes evident in the results. Our model achieved a score of 0.716 as Matthew's Correlation Coefficient and a balanced accuracy of 0.84. It clearly outperforms our baseline and competes with renowned existing signal peptide prediction methods.

Availability: The datasets used for training and testing were kindly made available by the makers of SignalP 5.0 and can be downloaded from http://www.cbs.dtu.dk/services/SignalP/data.php.

Contact: [ ]

Supplementary information: Supplementary data are included in the appendix.

1 Introduction

Signal peptides (SPs) are cleavable amino acid sequences that are commonly located at the N-terminal region of newly nascent secretory proteins. Their main task is to direct their individual protein into or across a cellular membrane before being removed during translocation through the membrane.

One of the common features which the majority of SPs share is the characteristic size of about 15 to 40 amino acids. SPs also all display a distinctive tripartite structure. It comprises an often positively charged n-region, which usually includes the first 1 to 5 amino acids after the initiator methionine and shows the most significant variation in length (Martoglio et al., 1998). Moving towards the C-terminal end of the protein, a longer stretch of hydrophobic amino acids is connected, known as the h-region that makes up the core of the signal peptide. The polar carboxyl-terminal c-region follows, before the signal peptide concludes with a characteristic motif, commonly featuring small, uncharged amino acids, such as alanine, at the ends. This sequence of 3 residues marks the signal peptide peptidase recognition site and is required for the cleavage to succeed (Nielsen et al., 1998).

Even though most SPs depict this three-domain-structure (von Heijne, 1985), SPs do vary in terms of actual sequence composition and length. As SPs have a stronger influence on protein biogenesis than initially expected, this diversity results in major differences in targeting pathway selection, translocation efficiency, as well as peptide cleavage and post-cleavage events (Hegde et al., 2006).

One of the most important export machineries for proteins among all domains of life is the general secretory pathway (Sec). This transport route offers two different modes of translocation (Fig. 1A), yet in both cases, the protein has to remain in an unfolded state. SPs stay attached to the rest of the sequence until or even after the protein passes the membrane. In the cotranslational manner, the signal peptide gets recognized by a ribonucleoprotein called signal recognition particle (SRP). This process happens already during protein synthesis, immediately after the nascent amino acid chain begins to leave the ribosome. The SRP-protein-ribosome complex moves to a SRP receptor located in the membrane, before then being transferred to the translocon, a membrane-bound protein complex. Protein synthesis continues directly through the translocon pore. This Sec mode proves to be preferred by proteins which swiftly lose the capability of being translocated due to fast aggregation in the cytosol, highly hydrophobic transmembrane proteins being an example. In the posttranslational manner, the protein is kept from folding as chaperones guide the protein to the translocation pore, directly after being released from the ribosome. The signal peptide then gets recognized by the translocon. Step by step, the protein gets pushed through the pore. This mode of the Sec export pathway is often favored in quickly growing organisms, as it produces a higher secretion rate due to translocation and the slow process of synthesis being uncoupled. SPs which block the SRP molecule due to a low level of hydrophobicity will take the posttranslational route as well (Freudl, 2018; Owji et al., 2018, Hegde et al., 2006).

The Sec pathway however is not the only existing export mechanism in cells. Characterized by the conserved consensus motif containing two arginine residues and occurring between the n-region and the h-region of SPs that are found using this route, it is called twin-arginine translocation pathway (Tat). Notably, the Tat machinery (Fig. 1B) only exists in Bacteria, plant chloroplasts and Archaea, and on the contrary to Sec, it transports its proteins in a fully folded conformation. After complete synthesis, the protein folds into its mature state, often involving the binding of cofactors. A membrane-bound receptor then recognizes the protein and recruits another protein complex, which transports the substrate across the membrane. The signal peptide gets cleaved off, while the final protein gets released onto the other side. Oppositely to the Sec pathway, which can be ATP dependent, the Tat route only makes use of the proton motif force (PMF). Adding to the Tat consensus motif, respective SPs share an overall greater length, especially because of the longer n-region. Their h-region also proves to be less hydrophobic. Positively charged amino acids in the c-region prevent the protein from being translocated through the Sec system (Freudl, 2018; Owji et al., 2018, Hegde et al., 2006).

SPs likewise influence protein biogenesis as variations in the peptide sequence lead to diversity amongst SPs in terms of translocation efficiency and interaction with the translocon. The latter interacts differently with each signal sequence in a substrate-dependent way. Dissimilar SPs may need other accessory factors for the recognition by the translocon to succeed, which even might not be immediately involved in the concrete interaction happening. A stable interaction between the signal peptide and the translocon results in the protein forming a looped

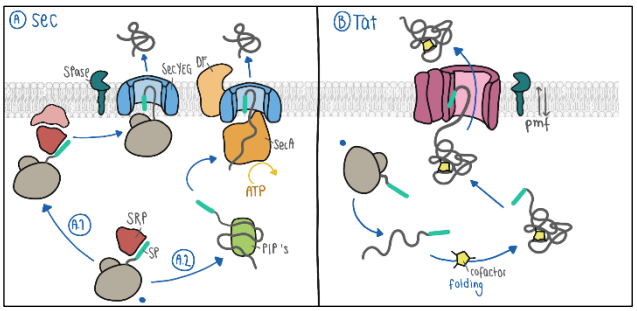

Fig. 1. The main export pathways. (A) The Sec pathway offers 2 different modes. (A.1) In the cotranslational mode, SRP recognizes SPs. Translocation is carried out during synthesis of the protein chain. (A.2) In the posttranslational mode, the ribosome finishes protein synthesis, but the product is still kept in an unfolded state by posttranslationally interacting proteins (PIPs). Secretion through the translocon happens step by step in an ATP-dependent manner. (B) The Tat pathway can involve the insertion of a cofactor, while the protein begins to adopt a folded conformation in the cytosol. Here, translocation occurs in a folded state using the PMF.

orientation with the N-terminal end still on the cytoplasmic side, the membrane channel opening and translocation beginning. However, if this interaction is not strong enough or even blocked by certain small molecules, the amino acid chain might slip back into the cytosol. The inefficiency of SPs may lead to a minor, yet detectable, population of the respective protein in the cytosol. In some cases, this even seems to be physiologically relevant (Hegde et al., 2006).

Given gating the translocon was successful, SPs can influence events occurring after the initiation of translocation as well. Similarity in gating efficiency still allows diversity in other aspects of translocation, for instance, folding of the protein after crossing the membrane or glycosylation and possible consequences (Hegde et al., 2006).

SPs even regulate the timing of their own cleavage. For cleavage to occur after a successfully completed gating process, the signal peptide peptidase recognition site, which we mentioned in the beginning, has to be present. Numerous features of SPs influence their removal, for instance, the length of the h-region that might appear inhibitory above a certain size. The availability of an uncharged residue at position -3 and a residue carrying a short side chain at positon -1, both respective to the cleavage site, seems to yet remain the necessary condition for cleavage (Martoglio et al., 1998). In terms of timing, the removal of SPs can happen in one or multiple steps, which may occur at different speeds. Depending on signal peptide composition, the process can ensue rapidly or take hours (Hegde et al., 2006).

Post-removal, most SPs immediately undergo further proteolysis in the membrane. The resulting fragments are then degraded by proteases after being liberated back into the cytoplasm. Some SPs however take on special post-cleavage roles, as part of the self-antigens shown on immune cells are recycled SPs. Generally speaking, the released signal peptide fragments could interact with any compatible protein in the cytosol (Martoglio et al, 1998).

Surprisingly, a fraction of SPs does not experience cleavage but remains attached to the rest of the protein chain after the latter got transported through the membrane. These so-called signal anchors carry a particularly long h-region that resembles an alpha-helix, which is characteristically found in transmembrane proteins. With the amino-terminal end of the chain remaining on the cytoplasmic side of the membrane, the anchored protein classifies as a type II transmembrane protein, referring to its orientation (Nielsen et al., 1998). Therefore the similarity between transmembrane helices, or signal anchors, and the hydrophobic regions of SPs is one of the major problems signal peptide prediction methods have to face.

SignalP 5.0 (Almagro Armenteros et al., 2019) is a bioinformatic tool that does outstandingly well at discriminating SPs and signal anchors and is known as state of the art. It manages to differentiate 3 signal peptide types across all domains of life and even outperforms methods that specialize in predicting the respective type. Its deep recurrent neural neural network construction joined with a conditional random field classification and additional optimized transfer learning portrays the complexity of this algorithm and the decades of profound research put into it.

Here, we present a much simpler built signal peptide prediction approach that was inspired by the great work behind SignalP 5.0. Unlike SignalP 5.0, our prediction is binary. For each residue, the model decides whether a specific residue belongs to a signal peptide. As this project emerged as part of a seminar, it was limited in time. Still, we achieved a functioning pipeline and presentable results.

2 Material and Methods

2.1 Dataset structure

The datasets used for the whole project are the training set and the test set available as FASTA files that were developed for training and benchmarking of SignalP 5.0. Both include protein sequences with a maximum length of 70 amino acids from Eukarya, Gram-positive Bacteria, Gram-negative Bacteria and Archaea. They either carry a signal peptide, an experimentally validated transmembrane (TM) section or neither. Distinguished types of SPs are Sec/SPI (SP), Sec/SPII (LIPO) and Tat/SPI (TAT), which refers to the used pathway and the type of signal peptide peptidase that performs peptide cleavage. Sec/SPII SPs are named "LIPO" as they carry a C-terminal "lipobox" motif. The training set of 20,758 proteins was clustered and homology-partitioned into 5 splits. To construct the benchmark set with its 8,811 resulting sequences, the training set underwent homology-reduction towards the training set of SignalP4, which was used by the most recently published method, DeepSig (Savojardo et al., 2017). Details concerning construction of the SignalP 5.0 datasets are available under 'Methods' in the paper presenting the tool (Almagro Armenteros et al., 2019).

The following numbers concerning the composition of the dataset all refer to the training set. However, analysis of the benchmark set shows that the distributions found in the latter do not deviate severely unless stated otherwise. Looking at the global dataset from a per residue perspective, "I" (cytoplasmic non-signal-peptide part of the sequence) appears to be the dominant label, making up 70% of all labels, followed by "O" (extracellular non-signal-peptide part) with 19%. This is no surprise: Most of the non-TM proteins are labelled as "I", while for those containing SPs, the rest of the sequence is labelled as "O", as SPs usually target their proteins to the extracellular or "outer" side of the membrane. In terms of domains of life, the dataset displays a clear bias towards Eukarya, representing 83% of all proteins. A possible reason could be the advanced biological relevance of eukaryotic sequences, but an official answer was not disclosed by the authors of SignalP 5.0.

Looking at the different domains individually, the distribution of labels varies a lot. In Eukarya for instance, neither "L" nor "T" appear, as both the Tat pathway and signal peptide peptidase II (SPII) do not exist in this domain of life. Also, the level of 81% "I" content is extremely high, which could be a result of the bigger portion of proteins without SPs, therefore increasing the number of "I"-labelled residues significantly. The global dominance of "I" labels becomes even more comprehensible, as 75% of

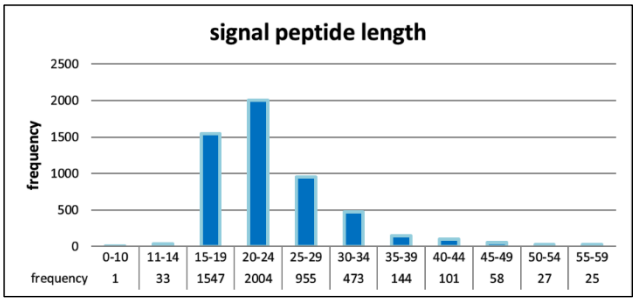

Fig. 2. Signal peptide lengths in the training set. Depending on signal peptide type, the average amino acid chain lengths vary. Globally, the average size of SPs is 23.6 residues.

all sequences do not carry a signal peptide, while only 16% (SP), 8% (TAT) and 2% (LIPO) do. Adding to this imbalance, only 21% (Gram-positive) or 18% (Gram-negative) of bacterial sequences do not bear SPs, yet in Eukarya, the signal peptide is absent in 85% of proteins.

The lengths of SPs (Fig. 2) vary depending on the respective type. SP-labelled proteins on average carry a signal peptide consisting of 23 amino acids, whereas LIPO proteins contain an average of 21 residues. TAT sequences however feature an average length of 36 amino acids because their h-regions are significantly longer, as mentioned before. This also supports the fact that in TAT proteins, 53% of all residues belong to SPs, while in other sequences it is only 34% (SP) or 30% (LIPO) of all residues.

2.2 Coding process and final model implementation

Starting off, we began by transforming the sequences given in the datasets into a suitable input format for our model. As the underlying principle of representing the individual residues, we chose one-hot encoding, because of its apparent structure and intuitive concept. In this case, one-hot encoding describes the creation of one zero vector of length 20 per residue. As it represents the 20 amino acids, one of the 20 fields is set to 1, which indicates the present residue. We expanded the resulting column vectors to sliding window matrices in order to include the surrounding amino acids for each residue, making it possible for the model to recognize motifs or regions. For this, we added 5 20-row-column vectors on the left and 5 on the right side of the currently regarded amino acid. The next feature we decided to include is the hydrophobicity of each residue in the sliding window, since this particular chemical property seems to influence protein characteristics significantly. Therefore, we added another row to the matrix. For measuring hydrophobicity, we settled with the Eisenberg scale (Eisenberg et al., 1984), because it showed the best results in comparison to alternatives, for instance the Kyte-Doolittle scales (Kyte and Doolittle, 1982). Similar reasons support the addition of polarity as a feature, as it is another effective way of highlighting the differences between the input residues. The normalized pKa-values of the respective α-NH3-group proved to be the optimal way of encoding polarity, compared to binary encoding and the non-normalized values, and were added as the next row to our matrix. Another manner of partitioning amino acids is using the electric charge. We decided to adapt a binary indication of positivity, negativity and neutrality, respectively, adding 3 rows to the matrix. The use of these chemical properties (hydrophobicity, polarity and electric charge) as features is strongly supported by the tripartite structure of SPs, as the domains bear residues that tend to show extreme values in one of these characteristics. Finally, the sliding window matrix should contain 11 column vectors of length 25. However, the emerging machine learning model requires an appropriately formatted input, so we flattened the matrix to a vector of length 275. As the final feature, we added the exact

position of the residue in the protein, beginning at the N-terminus, divided by the sequence length as one single value to the flatted matrix.

For the actual model, the majority of instances we used are provided by Sci-kit Learn (Pedregosa et al., 2018), as are the multiple estimators we tested. After showing better results than the alternatives, being the Support Vector Classifier (SVC) and the Random Forest Classifier (RFC), we chose an artificial neural network in the form of the Multilayer Perceptron (MLP). The parameters of our final model are an adaptive learning rate and a maximum of 500 iterations. As the single hyperparameter to optimize, we decided on the hidden layer size. The option that was selected the most often turned out to be a single hidden layer containing 130 nodes. For scoring and as the metric for optimization, we chose the Matthew's Correlation Coefficient (MCC) (Matthews, 1975), which will be explained in detail later in 2.3 ("Assessment and evaluation"). In short, the MCC is generally regarded as a good measure to represent the confusion matrix in a single score and known to be resistant against data imbalance. This resistance may be very suitable in our case.

Another essential part of our machine learning pipeline is the manually implemented cross-validation. Although Sci-kit Learn offers efficient cross-validation frameworks such as the Stratified K-Fold object, we resolved a manual implementation, as the data contained in the FASTA files was already partitioned into 5 splits per default, which were not chosen randomly but according to homology. Therefore it is important to respect these predefined splits, in order to avoid falsifying scores by training on proteins similar to the test samples. We are still using the Sci-kit learn GridSearchCV object that enables us to combine cross-validation and hyperparameter optimization in an efficient way, but we are manually assembling the indices needed for the creation of splits. To ensure a maximized use of the whole dataset, we expanded the simple cross-validation by fitting it into another cross-validation, forming a nested constellation. One of the 5 rounds of the outer cross-validation includes 4 rotations in the inner one, involving training on 3 splits and validating on 1, all taken out of the training set. Afterwards the model gets tested on the remaining split, taken out of the benchmark set.

We soon observed the need for imbalance correction when looking at the datasets, as the ratio of the minority class (SP-labelled residues) to the majority class (NO_SP-labelled residues) is 0.09. Instead of manually resampling the data, we were looking for a framework that would make it possible for us to combine cross-validation and imbalance correction. The pipeline object of Imbalanced-learn (Lemaitre et al., 2016) enables different steps of sampling and estimating to occur sequentially and independently. Our next step was to choose a resampling technique. We started off with SMOTE (Chawla et al., 2002), a popular technique of oversampling the minority class. Instead of classic oversampling with replacement of residues, SMOTE creates real artificial examples. Though, when only trying for a 0.12 ratio while letting the algorithm run on the Rechnerhalle server of the Technical University of Munich, it took several days to complete a single rotation of cross-validation including taking up most of the available resources. Next, we tested random undersampling of the majority class, which is possible in our case thanks to the extensive training set, as well as randomly oversampling the minority class. We finally settled with the latter, as training scores improved significantly while consuming reasonable time and space. The final ratio of minority to majority class we aimed for is 0.25, in comparison to the original 0.09.

2.3 Assessment and evaluation

Sci-kit Learn offers multiple common statistical measurements. For the assessment of our prediction quality, we use the MCC, accuracy, precision, recall, the area under (AUC) the receiver operating characteristic (ROC) curve. We added balanced accuracy, because like the MCC, it is noted as rather resistant against imbalanced data. The metrics are defined as following:

FN = false negatives, FP = false positives,
TP = true positives, TN = true negatives

A) Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ B) Specificity = $\frac{TN}{TN+FP}$

C) Recall = $\frac{TP}{TP+FN}$ = True-Positive-Rate (TPR)

D) MCC = $\frac{TP\ x\ TN-FP\ x\ FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$ E) Precision = $\frac{TP}{TP+FP}$

F) Balanced accuracy = $\frac{Recall+Specificty}{2}$ G) FP-Rate (FPR) = $\frac{FP}{FP+TN}$

Specificity (Equation B) can be described as a measure of the ability of the model of avoiding falsely predicting SPs. Recall (C) gives information about how good the model is at detecting the true SPs. Precision (E) represents the number of predicted SPs that truly were SPs, compared to the number of all positive predictions. Accuracy (A) displays the fraction of rightly identified residues divided by the total number of residues. The FPR (G) is a measure of describing how bad the model is at predicting the residues that do not belong to a signal peptide correctly.

Additional to the performance metrics above, we provide a confusion matrix (Fig. 3, Supplementary Figure 3). We also included a ROC-curve, presenting the relation between recall and the FPR regarding different cutoffs, as well as a precision-recall curve. For plotting we made use of the Matplotlib package (Hunter, 2007).

To show the superiority of the final machine learning model we compared it to a baseline prediction for which used the naïve informed random approach. Its performance should be surpassed by our model. To create a biased dice that chooses one label per amino acid according to given probabilities, we made use of the global distribution of residues.

3 Results and Discussion

3.1 Final model evaluation

Our model fulfilled all minimum requirements by achieving higher scores across all performance measurements, compared to the baseline values (Supplementary Table 1). With an MCC of -0.0002, our baseline seems to be reasonably close to the expected scores of a random predictor.

When looking at the receiver operating characteristic (ROC) curve (Supplementary Figure 1) with an AUC value of 0.973, we may get the impression of an exceptionally well performing classifier. Here, it is worth computing the precision-recall curve (Supplementary Figure 2), which looks promising as well, but in a rather restrained manner. On the contrary to the ROC curve, this precision-recall curve does not make use of the false-positive rate (FPR), which takes the number of TNs into account. Due to the present dataset imbalance, this value is disproportionally high and may therefore distort resulting computations such as the ROC curve.

This distortion becomes evident when looking at the confusion matrix of our final model (Fig. 3). The coloring that was chosen in proportion to the absolute values in the table already gives evidence of a large number of true negatives in comparison to the others. Unsurprisingly, the number of FNs is the smallest, as the model is very unlikely to falsely predict one of the many NO_SP labeled residues as part of a signal peptide.

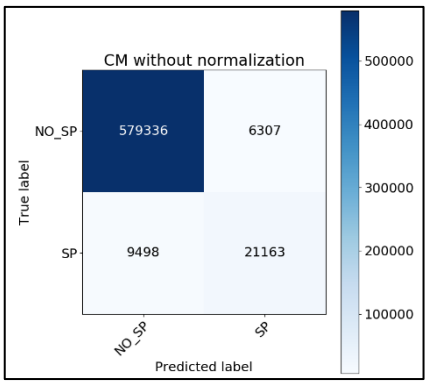

Fig. 3. Non-normalized Confusion matrix of the final model. The coloring as well as the absolute values of true positives, false positives, false negatives and true negatives suggest an unproportionally high number of the latter. The consequences of an existing imbalance become evident.

When looking at the normalized confusion matrix (Supplementary Figure 3), we observe that our model performs well in terms of predicting the amino acids not belonging to a signal peptide, labelling 99% of these residues correctly. For the biologically more relevant task, being the prediction of signal-peptide associated residues, our model properly identifies the residue in 69% of cases. The hope for improvement if the dataset contained as many positive (SP) residues to train on as there are negatives (NO_SP) is part of what lead us to trying imbalance correction.

3.2 Final model evaluation

Based on the distribution of labels in the dataset, we assumed imbalance correction would not only be a necessary, but also a beneficial step. However, not resampling the data during training turned out to result in the best training and performance scores (Supplementary Table 2). The final model achieves a MCC score of 0.718 as the mean of the inner cross-validation training scores, being superior to several alternatives based on random oversampling or undersampling. These values, as well as the prediction accuracy and the AUC of the ROC curve do not show high variations, revolving around 0.96 and 0.97, respectively. Other measurements on the other hand do show a certain trend: As the desired ratio of minority to majority class rises and the imbalance declines, the prediction precision values decrease, while the prediction recall values increase. This development can be explained by observing the definitions. Instead of the number of FNs, like recall does, precision uses the number of FPs. In comparison to a model based on an oversampled dataset, the final model trained on less positively labeled amino acids, and is therefore less likely to falsely predict a positive residue. Thus, the number of FPs decreases and the precision score rises, the lesser the majority to minority class ratio. By raising this ratio the model becomes less likely to label residues as 0, the number of FNs decreases and the recall scores increase. As recall influences the value of balanced accuracy, the latter shows an analogue development. Against our assumption, the MCC scores lower as imbalance correction grows. Possible reasons for this might be a loss of data, regarding undersampling, or the model adapting to certain outliers due to oversampling of the latter.

Manual cross-validation proofs to be the right choice (Supplementary Table 3). Our manual implementation according the given dataset splits shows better values in all measurements. Notably, the discrepancy between the mean of the inner validation scores and the outer prediction MCC is minimal compared to the difference found in the other model. Therefore it is beneficial to avoid replication of redundancy during training by implementing cross-validation manually.

3.3 Comparison to state of the art: SignalP 5.0

SignalP 5.0 and our model unfortunately are not easily comparable. First, SignalP 5.0 uses certain parts of the dataset as the positive dataset, as the authors provide scores for the detection of specific labels, yet not for a binary prediction. Second, SignalP 5.0 presents individual scores for each domain instead of an overall assessment. We aimed for a similar domain-specific presentation of our predictions, but it was not possible to make these scores available. This may be due to an unbalanced domain-specific confusion matrix, missing either TPs, FPs or FNs, and therefore inhibiting performance metric calculations.

Our model achieved an MCC of 0.716 during benchmarking, with a positive dataset of all signal peptide types while transmembrane and globular proteins make up the negative dataset. SignalP 5.0 accomplishes MCCs of 0.86 for Gram-negative bacteria, up to 0.922 for Archaea, when the positive dataset is comprised of Sec/SPI labeled proteins only. Although the state of the art clearly outperforms our model, the latter can still compete with other available signal peptide prediction algorithms. Signal-CF (Chou et al., 2007) and SOSUIsignal (Gomi et al., 2004), as presented in Supplementary Table 7 of the SignalP 5.0 publication (Almagro Armenteros et al., 2019), perform similar to our model across all domains.

This comparison of results becomes even more difficult when looking at other types of SPs comprising the positive dataset of the predictions disclosed in the publication of SignalP 5.0. Sec/SPII and Tat/SPI signal peptide carrying proteins make up a minority of proteins across the whole dataset as well as among SPs. Although our MCC of 0.716 is within the range of the scores of PRED-LIPO (Bagos et al., 2008) and TatP for Sec/SPII and Tat/SPI prediction respectively, it is not reliable to state an equal performance of our model.

Our machine learning pipeline is already achieving presentable per residue prediction results, confidently detecting differences between amino acids belonging to SPs and those amongst other parts of the sequence. To capture true biological relevance, the next step on our way to a publishable signal peptide prediction algorithm would be predicting at a per-protein level. Like SignalP 5.0, the classification of whole SPs and their cleavage sites could yield first biologically relevant results. Upgrading to label-specific instead of a binary prediction would be another possible way of expanding our model.

4 Conclusion

SPs are an important subject to current research in the production of recombinant proteins. Computationally predicting SPs can improve the efficiency of the process. Utilizing their characteristic structure, we built a per-residue classifier which achieves an MCC prediction score of 0.716. Although the dataset displays a high level of imbalance, methods for correcting the latter did not result in better results. However, our manual implementation of a nested cross-validation outperforms versions of our model which do not respect the given dataset partitions and therefore increase redundancy. Further possible improvements are upgrading the binary classification to the prediction of specific types of SPs or predicting the cleavage site. While being outperformed by the sophisticated state of the art SignalP 5.0 algorithm, our model competes well within the range of other available signal peptide predictors

Acknowledgements

Thanks to My Supervisor for continuous support and advice as the main supervisor of the project. Lastly, thanks to all other participants in our Problem-Based Learning Seminar for the helpful discussions after presenting in class.

Conflict of Interest: none declared.

References

Almagro Armenteros, José Juan, et al. “SignalP 5.0 Improves Signal Peptide Predictions Using Deep Neural Networks.” *Nature Biotechnology*, vol. 37, no. 4, Apr. 2019, pp. 420–23. *DOI.org (Crossref)*, doi:10.1038/s41587-019-0036-z.

Bagos, Pantelis G., et al. “Prediction of Lipoprotein Signal Peptides in Gram-Positive Bacteria with a Hidden Markov Model.” *Journal of Proteome Research*, vol. 7, no. 12, Dec. 2008, pp. 5082–93. *DOI.org (Crossref)*, doi:10.1021/pr800162c.

Bendtsen, Jannick, et al. “Prediction of twin-arginine signal peptides.” *BMC Bioinformatics*, vol. 6, no. 1, 2005, p. 167. *DOI.org (Crossref)*, doi:10.1186/1471-2105-6-167.

Chawla, N. V., et al. “SMOTE: Synthetic Minority Over-Sampling Technique.” *Journal of Artificial Intelligence Research*, vol. 16, June 2002, pp. 321–57. *arXiv.org*, doi:10.1613/jair.953.

Chou, Kuo-Chen, and Hong-Bin Shen. “Signal-CF: A Subsite-Coupled and Window-Fusing Approach for Predicting Signal Peptides.” *Biochemical and Biophysical Research Communications*, vol. 357, no. 3, June 2007, pp. 633–40. *DOI.org (Crossref)*, doi:10.1016/j.bbrc.2007.03.162.

Eisenberg, D., et al. “Analysis of Membrane and Surface Protein Sequences with the Hydrophobic Moment Plot.” *Journal of Molecular Biology*, vol. 179, no. 1, Oct. 1984, pp. 125–42. *DOI.org (Crossref)*, doi:10.1016/0022-2836(84)90309-7.

Freudl, Roland. “Signal Peptides for Recombinant Protein Secretion in Bacterial Expression Systems.” *Microbial Cell Factories*, vol. 17, no. 1, Dec. 2018, p. 52. *DOI.org (Crossref)*, doi:10.1186/s12934-018-0901-3.

Gomi, Masahiro, et al. “High Performance System for Signal Peptide Prediction: SOSUIsignal.” *Chem-Bio Informatics Journal*, vol. 4, no. 4, 2004, pp. 142–47. *DOI.org (Crossref)*, doi:10.1273/cbij.4.142.

Hegde, Ramanujan S., and Harris D. Bernstein. “The Surprising Complexity of Signal Sequences.” *Trends in Biochemical Sciences*, vol. 31, no. 10, Oct. 2006, pp. 563–71. *DOI.org (Crossref)*, doi:10.1016/j.tibs.2006.08.004.

Hunter, John D. “Matplotlib: A 2D Graphics Environment.” *Computing in Science & Engineering*, vol. 9, no. 3, 2007, pp. 90–95. *DOI.org (Crossref)*, doi:10.1109/MCSE.2007.55.

Kyte, Jack, and Russell F. Doolittle. “A Simple Method for Displaying the Hydropathic Character of a Protein.” *Journal of Molecular Biology*, vol. 157, no. 1, May 1982, pp. 105–32. *DOI.org (Crossref)*, doi:10.1016/0022-2836(82)90515-0.

Lemaitre, Guillaume, et al. “Imbalanced-Learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning.” *ArXiv:1609.06570 [Cs]*, Sept. 2016. *arXiv.org*, http://arxiv.org/abs/1609.06570.

Martoglio, Bruno, and Bernhard Dobberstein. “Signal Sequences: More than Just Greasy Peptides.” *Trends in Cell Biology*, vol. 8, no. 10, Oct. 1998, pp. 410–15. *DOI.org (Crossref)*, doi:10.1016/S0962-8924(98)01360-9.

Matthews, B. W. “Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme.” *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, Oct. 1975, pp. 442–51. *DOI.org (Crossref)*, doi:10.1016/0005-2795(75)90109-9.

Nielsen, H. “From Sequence to Sorting. Prediction of Signal Peptides”. *PhD Dissertation (Stockholm, Sweden: Stockholm University)*, 1999.

Nielsen, H., and A. Krogh. “Prediction of Signal Peptides and Signal Anchors by a Hidden Markov Model.” *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, vol. 6, 1998, pp. 122–30.

Owji, Hajar, et al. “A Comprehensive Review of Signal Peptides: Structure, Roles, and Applications.” *European Journal of Cell Biology*, vol. 97, no. 6, Aug. 2018, pp. 422–41. *DOI.org (Crossref)*, doi:10.1016/j.ejcb.2018.06.003.

Pedregosa, Fabian, et al. “Scikit-Learn: Machine Learning in Python.” *ArXiv:1201.0490 [Cs]*, June 2018. *arXiv.org*, http://arxiv.org/abs/1201.0490.

Savojardo, Castrense, et al. “DeepSig: Deep Learning Improves Signal Peptide Detection in Proteins.” *Bioinformatics*, edited by Alfonso Valencia, vol. 34, no. 10, May 2018, pp. 1690–96. *DOI.org (Crossref)*, doi:10.1093/bioinformatics/btx818.

von Heijne, Gunnar. “Signal Sequences.” *Journal of Molecular Biology*, vol. 184, no. 1, July 1985, pp. 99–105. *DOI.org (Crossref)*, doi:10.1016/0022-2836(85)90046-4.