KTH Royal Institute oF Technology

DD2404

Applied Bioinformatics

6

Predicting signal peptides

Authors:
J. Matak, J. Mrđen

December 27, 2018

Contents

1 Abstract

Signal peptides offer various insights into genomics and proteomics investigation, therefore their prediction can benefit several studies closely related to discovering drugs, vaccines and diagnosing diseases. This paper is based on applying machine learning techniques to determine whether signal peptide exists in protein sequence, once one is provided.

To address this problem, a neural network based approach is used. Architecture was designed along with method of feature extraction from protein sequence to successfully predict signal peptides. Training data was extracted from previous study, and given in form of separate datasets with positive and negative samples. Furthermore, problem was observed from perspective of predicting signal peptides of different protein types, transmembrane and non-transmembrane.

Final results were satisfiable. In each test, more than 82% proteins were properly classified. Same model was used on human and pig's proteomes, resulting in less than 10% of proteins being classified containing signal peptides.

2 Introduction

A signal peptide is a short peptide (about 16-30 amino acids long) usually located at the N-terminus of the synthesized proteins. It serves as a postal code to give the cell information where to translocate the protein. The process is also called as protein secretion and it is mostly done across the secretory pathway(7). The interesting part is that all the information where the protein is carried lies in the structure of the signal peptide.

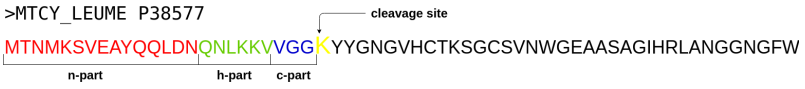

Figure 1: Signal peptide divided in three parts, n,h and c

The structure itself of the signal peptide consists of 3 parts (Figure 2.). The first part is called the N part which tends to be positively charged. Followed by that,

the h-region is present which usually forms a single alpha-helix. In addition, there is a c-part, usually polar but not charged, at whose last part the signal sequence is cut out from the protein after successful secretion process. The enzyme for cleaving off the signal sequence was therefore named with the signal peptidase. Since the pattern in majority of cases consists of those 3 parts, one can build a predicting machine in order to localize the signal peptides inside a protein. The difficulty lies in the degree of sequence conservation, as well as length. Lengths of named parts in given database are visualized in Figure 2.

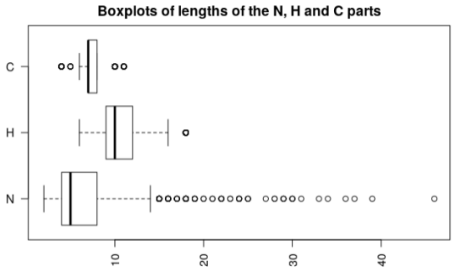

Figure 2: Boxplot exposing lenghts of N,H and C part in given database of proteins.

The applications of signal peptide classifiers are used vastly in genomics and for proteomics investigation. For example, some secreted proteins play an important part in the pathogenicity of bacteria, and therefore the results of predictions can be used to develop useful drugs, vaccines and diagnostics (3). Moreover, investigations with signal peptides are also used for reprogramming cells for gene therapy.

In this paper, we have developed a signal peptide classifier based on a neural network approach. The results can be seen in the following pages of the paper, as well as explanations of implementation and used technologies.

3 Methods

We took a neural network based learning approach with the sliding window technique as a basis for building the signal peptide classifier. The things and problems to consider while creating the classifier was coding of the protein sequences, designing the network architecture, and at the end the classification principle of unknown examples.

3.1 Learning data and preprocessing

The data provided for building the classifier was divided into 2 parts: transmembrane and non-transmembrane proteins. Files which contained the protein structures were in Fasta-like format, with an additional annotation line serving as a reference to the location of the signal peptide. Letters 'n', 'h', 'c' and 'o' would mark the part or the absence of the signal peptide. The last amino acid of a signal peptide was annotated with a capital C, showing the cleavage site.

Since protein sequences consist of amino acids that are in biological world represented as capital letters, we used one-hot encoding to represent each amino acid as its binary representation. Since there are 20 amino acids, each amino acid was coded as an array of 19 zeroes and a '1' at its unique place for an amino acid.

Protein sequences can also be of varying length, as well as signal peptides. As a neural network input must be fixed, we used a sliding window to go over all protein sequences and capture each sequence with a fixed length and an offset from beginning (1). That way we augmented our data as we could extract multiple training samples from one protein sequence. All the captured samples were used as a training data for the neural network.

The label of each such training sample was binary 2 dimensional. First value suggested whether the signal peptide was localized in the sliding window, and it had a value of 1 if either one of the 3 parts of the signal peptide was shown in the annotated output of the captured sequence. The second value had a value of 1 if a cleavage site was present at the captured sequence. The purpose of such mapping of the output was to design a classifier which could combine the knowledge of the given information about the parts of the signal peptides and the cleavage site. Moreover, such classifier could be used for plotting the confidence if a signal peptide was present at a certain place or not.

3.2 Learning approach

For training itself, a feed-forward neural network was proposed. The input size of the neural network is the size of the sliding window multiplied by the size of each coded amino acid. The architecture of the neural network can be defined by the user with the program arguments of the developed software. The output of the neural network is binary 2 dimensional, described in the subsection above.

Such neural network is not a classifier by itself as it only extracts the information given from the amino acids. Therefore, we pipelined the outputs to another mecha-

nism for predicting if a signal peptide is actually present or not.

In the sequence of outputs, we would look for all occurrences of the cleavage site with the confidence of at least 20% or 0.2 . The reason for such small confidence is that the cleavage site is harder to notice than the signal peptide itself. Each time such confidence was located, the outputs from the beginning to the location place would be scanned, and if 90% of the outputs diagnosed that a signal peptide is present until the cleavage site was encountered, the mechanism would output positive presence of a signal peptide. For faster performances, during the evaluation of the unknown samples, only first 50 amino acids would be taken into account, because a signal peptide is always located at the beginning of the sequence.

4 Results

4.1 Training results

The data provided was preprocessed by dividing amino-acids in coding window with outputs coded to represent signal peptide and coding site. Followed by that, data was trained on roughly 580000 examples (coding window of size 25), each one coded independently. The results are shown in Table 1. We can see the tendency of improving precision until the point where window size was 27, which was above the average size of a signal peptide.

	PROTEIN CLASSIFICATION	*SIGNAL & CLEAVAGE PREDICTION*
Window size	Precision [whole dataset]	MSE [test set]
17	$61.90\% \pm 5.43\%$	$(7.81 \pm 0.636) \cdot 10^{-3}$
21	$77.40\% \pm 2.07\%$	$(6.39 \pm 0.153) \cdot 10^{-3}$
23	$78.35\% \pm 1.98\%$	$(5.89 \pm 0.266) \cdot 10^{-3}$
25	$84.89\% \pm 3.15\%$	$(6.01 \pm 0.487) \cdot 10^{-3}$
27	$81.38\% \pm 1.64\%$	$(4.86 \pm 0.384) \cdot 10^{-3}$

Table 1: Signal peptide classifier accuracy and mean square error on the neural network

Additionally, we split the data into non-transmembrane protein sequences and transmembrane protein sequences in order to find out which one are more difficult to classify. Results are shown in Table 2. We can see an obvious difference as it was harder for the model to distinguish whether a signal peptide was present in a non transmembrane protein.

Window size	TM protein accuracy	Non-TM protein accuracy
21	$91.44\% \pm 6.38\%$	$74.68\% \pm 8.50\%$
23	$92.98\% \pm 4.29\%$	$80.29\% \pm 6.02\%$
25	$93.84\% \pm 5.94\%$	$81.73\% \pm 4.06\%$
27	$94.51\% \pm 0.99\%$	$81.90\% \pm 7.30\%$

Table 2: Classifier accuracy percentage on transmembrane vs. non transmembrane protein sequences

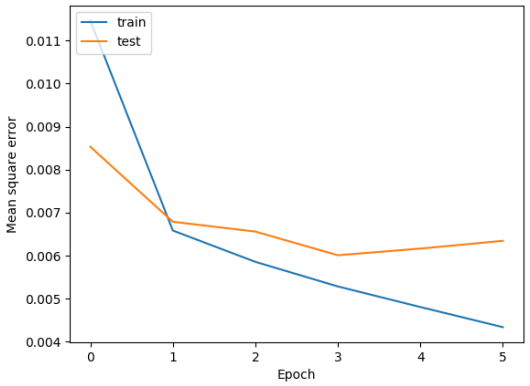

Figure 3: Training history showing mean square error on test and validation set

4.2 Proteome evaluation results

For further evaluation, we downloaded a human proteome and a wild boar proteome (6) and performed the classifier on it. For the human proteome, 1,464 proteins were classified as signal peptides out of 21,080, which is around nešto 6.94%, while wild boar had 1209 proteins out of 23229 (5.20 %).

4.3 Used technologies

Program for predicting signal peptides was built in Python version 3.6 using different modules. Feature extraction and data processing has been done in Numpy module, while Matplotlib was useful for draw diagrams of learning process and visualizing prediction results. All machine learning techniques which mostly include working with neural networks were done using Keras, API built on top of Tensorflow library.

Techniques used in this paper were not only related to programming, but for data and work organization. Model for such work was paper from William Stafford Noble which explains how to organize one bioinformatics project (4). Some of ideas were

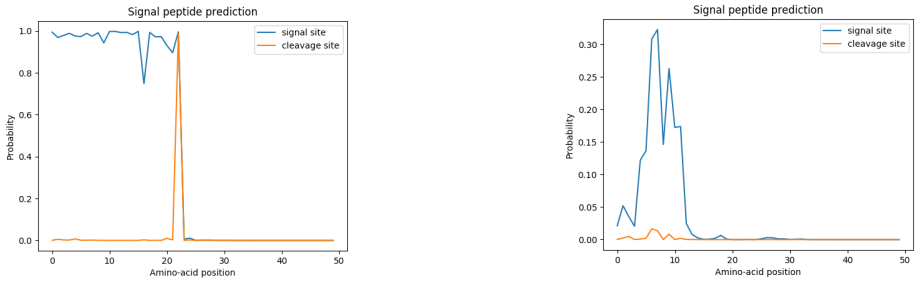

Figure 4: Left figure is showing typical example of prediction on signal peptide, while right side shows signal without cleavage site predicted, which results in classification as non containing signal peptide.

directly implemented as it was case with project structure, and keeping track of files, while there was addition from our side on programming side, where choice was using rather modulated code than collecting multiple outputs with different scripts.

5 Acknowledgments

The precisions we got for different window sizes were according to our assumptions, with larger window sizes getting more accurate precisions cause they focused on a bigger area of a protein and could interpret all the correlations in the connected amino acids. We were a bit sceptic about whether the NN would extract useful information for classification, but the results didn't turn out to be bad at all.

For the possible future work on this project, one could use a different model of neural networks which deals specifically with data sequences. By that, we strongly suggest using some variant of a recurrent neural network (RNN) or a long short-term memory network (LSTM network). Another thing one could look at is the coding technique of the sequences, as the one-hot encoding isn't the only option that can be used. For example, sparse and distributed encoding could prove to be a better solution.

Since the signal peptide is always located in the beginning of the sequence, one can cut out the last part of the protein structure cause it is not relevant and just train on the cropped sequences. Finally, there are always other machine learning methods that are not based on neural networks, but can be as efficient in predicting (5) (2).

References

[1] Soren Brunak Henrik Nielsen, Jacob Engelbrecht. *A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites*. 1997.

[2] Shoba Ranganathan Khar Heng Choo, Tin Wee Tan. *A comprehensive assessment of N-terminal signal peptides prediction methods*. 2009.

[3] Hiwa Målen Swati Prasad Inge Jonassen Nils Anders Leversen, Gustavo A. de Souza and Harald G. Wiker. *Evaluation of signal peptide prediction algorithms for identification of mycobacterial signal peptides using sequence data from proteomic methods*. 2009.

[4] William Stafford Noble. *A quick guide to organising computational biology projects*. 2009.

[5] Michael E. Riffle Jeff A. Bilmes William Stafford Noble Sheila M. Reynolds, Lukas Käll. *Transmembrane Topology and Signal Peptide Prediction Using Dynamic Bayesian Networks*. 2008.

[6] Uniprot. Proteome database. URL: `https://www.uniprot.org/proteomes/`.

[7] Wikipedia. Signal peptide. URL: `https://en.wikipedia.org/wiki/Signal_peptide`.