30 09 2019

Machine learning

Signal peptide prediction with SecVec embeddings

Thomas Heinzinger *

*To whom correspondence should be addressed.

Associate Editor: Michael Bernhofer

Abstract

Motivation: **S**ignal **p**eptides (sp) are essential for protein allocation within and outside of the cell for every known organism. A lot of drugs target them out of this reason and determining sps has become a more and more important prediction task in research and the pharmaceutical industry to reduce money and time of drug research. First signal peptide and generally protein prediction methods in the field used basic properties such as hydrogen bonds and residue charge. These methods have been continuously optimized until they hit an impossible hurdle. The key to overcome this hurdle was introduced by using evolutionary information out of protein alignments which nowadays almost every proficient approach utilizes. However, through new mass sequencing techniques protein databases are growing so fast, that alignments between proteins are a time intensive task even for highly optimized algorithms. The developers of SecVec[1] present a "deep-learning" solution as the next innovation in this field, whereby so called "**l**ong-**s**hort **t**erm **m**emory" (lstm) models are key for further improvements. These learn and capture different aspects of unlabeled and labeled proteins, which neither implicitly nor explicitly contain evolutionary information, on their own and provide a quick way to retrieve data for the prediction of different protein properties. In the publication of SecVec, they showed the functionality of the embeddings in prediction of secondary structure. Here we used it to find signal peptides in complete protein sequences and compare its performance to renowned methods such as SignalP5.0[2].
Results: We proposed a novel way to use the embeddings created by SeqVec to predict other protein features as structure and specifically signal peptides. We used a simple **C**onvolutional **N**eural **N**etwork (cnn) in combination with a **C**onditional **R**andom **F**ield (crf) to capture implicit information in the embeddings and to predicted four different types (classes) of signal peptides in different organisms. At a residue level we achieved an overall accuracy (Q4) of the types of 99.0% +- 0.1. Since the used data inherently comes with class imbalance (sp residues are less likely to appear then non sp ones) a measurement to assess the quality of the predictions per class, the **M**atthews **C**orrelation **C**oefficient (mcc), was utilized and lays around 0.893 +- 0.006. On a global, per protein level of signal peptide prediction (Does the protein contain a signal peptide and which type?) we attained a Q4 of 97.6% +- 0.2 and mcc of 0.906+-0.006 which means that overall more predictions were false but more predictions per class (signal peptide) right. Although, we cannot compete with the results of the developers of SignalP5.0 who achieved an overall mcc of around 0.94, we were able to show that evolutionary information is not a necessity anymore. Our model is held very simplistic which proves that SecVec is effectively capturing the biophysical properties of proteins and that these can be used for diverse predictions tasks.
Availability: https://github.com/tomthun/Masterpraktikum
Contact: heinzinger.thomas@gmail.com
Supplementary information: Supplementary data and information are included.

1 Introduction

Protein allocation within and outside of cells are important and complex biological tasks, often guided specific signal peptides (sp) contained in the N-terminus of protein sequences. Sps are usually around 16-30 amino acids long and used by every organism, including Archaea, Eukarya and Bacteria. Regarding the general secretory pathway, the organisms use different approaches whereby protein translocation in prokaryotes is directed across the plasma membrane and the endoplasmic reticulum membrane in eukaryotes. Therefore, proteins endowed with a signal

peptide are resident in the endoplasmic reticulum and Golgi apparatus, secreted proteins and proteins inserted in plasma membranes. In sum, protein destination and function are intertwined with their according signal peptides, which is why it is key to find reliable methods to predict these. Another main parameter to predict is the position at where the signal protein is cleaved off from its host protein, which is also known as the cleavage site. This happens during or after membrane translocation by digestion through signal peptidases. Over the last decade evolutionary information (e.g. evolutionary couplings[3]) have become most popular and efficient as fundamental data for prediction tools. Studying a protein over time yields valuable insights on how e.g. mutations (insertions, deletion, etc.) have effects on protein properties (BLOSUM[4]).

One of the first publicly available methods for signal peptide prediction is SignalP[5]. It has been continuously improved over the last decade: Frist, simple artificial neural networks were used for prediction[5], in the later versions hidden Markov models[6] and more complex deep learning architectures[7,2] were applied, resulting in improved cleavage site predictions and discrimination of signal peptides and transmembrane helices.

In its latest revision, the so called SignalP5.0[2], the authors also provided a publicly available dataset over 20758 proteins. The proteins in the dataset are annotated in a 3-line FASTA format, where three distinct signal peptides are being distinguished:

- Sec/SPI: "standard" secretory signal peptides transported by the Sec translocon and cleaved by Signal Peptidase I (Lep) and are exclusive to eukaryotes
- Sec/SPII: lipoprotein signal peptides transported by the Sec translocon and cleaved by Signal Peptidase II (Lsp)
- Tat/SPI: Tat signal peptides transported by the Tat translocon and cleaved by Signal Peptidase I (Lep)

Here, we utilized the dataset to predict these signal peptides with another new deep-learning method for protein properties called SecVec[1]. Originally used for protein folding and structure prediction its architecture allows for overarching usage regarding proteins. So-called long short-term memory networks[8] (LSTMs) process the information of protein properties into continuous vectors (embeddings). First, this idea arose in the field of natural language processing[9] since syntax and semantics of language can be learned by the probability distribution of words in sentences. Depended on the context of a sentence, words are then differently parameterized. This is advantageous since two identical words with possible different meanings can be contextually distinguished based on the composition of the sentence in which they are used. LSTMs can be similarly applied to proteins whereby proteins are seen as "sentences" and amino acids as "words". The models are fed with databases of proteins (e.g. Uniref50) in order to learn different sentences/protein compositions. Thereby no evolutionary information is used implicitly or explicitly.

A notable improvement is the speed at which embeddings can be created. Once a LSTM model is fully trained (note that this consumes most of the time) creating embeddings takes about 0.03 seconds[1]. Compared to that, most commonly methods are built around evolutionary information and couplings[10,11] by alignment of similar proteins. However, those algorithms are becoming increasingly computationally costly since the number of UniProt entries grow faster every year through next generation sequencing[12] methods. Even fast and highly optimized algorithms such as the HHblits3[13] need several minutes for finding and aligning similar proteins. Furthermore, evolutionary information is still missing for proteins in UniProt, e.g. the entire "Dark Proteome"[14] which consist of less-well studied proteins although they are important for function[15].

As part of this work we transformed the FASTA-dataset provided by SignalP5.0 into its vector representation using SecVec. Then we used the resulting embeddings and trained a simple one-layer convolutional neural network (cnn) with the addition of a conditional random field. Next, to assess the predictive power of the embeddings we distinguished between two levels: per-residue (word-level) and per-protein (sentence-level). Regarding per-residue level, we predicted three different signal peptides equally as in the original publication of SignalP5.0. Non-signal peptides can be differentiated between inner, outer and trans-membrane, but are merged into '*Others*'. On per-protein level, we simply observed if the per-residue prediction contains a signal peptide and if so, label the protein with the according sp type. Finally, to benchmark the efficiency of our method we used an according benchmark-dataset provided by SignalP5.0. This dataset a subset of proteins of the complete dataset whereby included ones are proteins that are less likely to be similar to others. Hence, if such can be detected with high likelihood the integrity of the network and comparison is ensured.

The results show that it is possible to use SecVec embeddings to predict distinct signal peptides down to residue level. In this project we reached an overall Matthews correlation coefficient (MCC[16]) of around 0.868 on the benchmark. Compared to that SignalP5.0 reached around 0.94. Although the latter achieved better results it is notable that here most simplistic architectures were utilized. Asides, it was possible to show that an abstract contextualization of proteins into vectors might inherit more information than a combination of protein properties and evolutionary information. The project successfully demonstrates the prediction of signal peptides and that

SecVec embeddings are not only applicable for learning protein structures.

2 Methods

Data: We trained and benchmarked a simple convolutional neural network on embeddings based on the FASTA files provided by the SignalP5.0 website. The datasets are publicly available under:
http://www.cbs.dtu.dk/services/SignalP/data.php
The embeddings have been created by applying the SecVec method. A detailed tutorial on how to use the method is given on the following GitHub page:
https://github.com/Rostlab/SeqVec
As stated in the methods section of SignalP5.0 the UniProt Knowledgebase release 2018_04[17] has been for data gathering. Further preprocessing of the data can be found in the publication (ref[2]). The training and benchmark datasets contain all 20 standard residue letters, 20758 and 8809 proteins respectively and every protein as well as contained residues are distinguished between three distinct signal peptides;

- Sec/SPI: "standard" secretory signal peptides transported by the Sec translocon and cleaved by Signal Peptidase I (Lep) with residue annotation **S**
- Sec/SPII: lipoprotein signal peptides transported by the Sec translocon and cleaved by Signal Peptidase II (Lsp) with residue annotation **L**
- Tat/SPI: Tat signal peptides transported by the Tat translocon and cleaved by Signal Peptidase I (Lep) with residue annotation **T**

as well as non-signal peptide annotations. Since non-signal peptide residues can be separately located in cytoplasm, transmembrane and extracellular, they are annotated with **I**, **M** and **O**. The header further contains information about animal kingdom (eukaryote, archaea, gram – positive/ negative), Uniprot identifiers and separation into specific protein subsets / "splits". The splits are relevant for cross-validation and observation of the learning progress of the CNN. Furthermore, one split will be excluded at the beginning of the training which is later extracted from the benchmark dataset and used for benchmarking purposes. The benchmark dataset has been created by 20% homology reduction with CD-Hit between the actual training set (SignalP5.0) and the one used for their latest published method Deep-Sig[18] (SignalP4.0). Thus, an independent dataset is created which they used to do unbiased comparisons of in house and other methods. Layout-wise, the benchmark dataset is the same as the training dataset.

At the current revision SeqVec produces an output vector of 1024*(protein length) of floating numbers that can range from negative to positive. The major reason for that is the network architecture of the algorithm which allows to capture multiple probable protein properties and report them in such a manner.

Data processing: Both datasets are processed in phyton version 3.7.3 (http://www.python.org) and learned/ evaluated with the help of the deep-learning framework pytorch[19]. Since proteins are generally around 466 and signal peptides respectively around 23 residues long, both datasets inherently have class imbalance between non signal and signal peptide residues. To strike this problem, first, proteins and embeddings have been shortened to a length of 70. In theory this should not lead to a loss of training information as SecVec was applied on the whole protein before. The information of residues at the far end of a protein is already be implicitly contained in the ones at the beginning. This indeed reduced class imbalance on a residue level, although data analysis revealed that there is still a strong disequilibrium as seen as in the following table:

Table 1. Relative distribution of signal peptide residue annotations across training and benchmark datasets.

	SIGNAL	TRAINING DATASET	BENCHMARK DATASET
NON	I	0.70	0.83
SP:	M	0.03	0.02
	O	0.18	0.10
SP:	S	0.05	0.02
	T	0.01	0.01
	L	0.02	0.02

A similar imbalance can be observed at a protein level where 0.74 for training and 0.86 percent for benchmark do not have a signal peptide. Secondly, we decided to address this disparity by weighting each class individually (class weighting is available in pytorch).
Moreover, 187 proteins that are shorter than 70 have been identified. Embeddings and labels of those have been masked accordingly (with no effect on learning) until they have the desired length. Equally to SignalP5.0 non-signal proteins have been summarized to one class before learning starts. In the following it will only be distinguished between non-SP and the three different kinds of SP.

Model architecture: For benchmarking purposes, we used the similar layout as described in SignalP5.0:

1. 2D convolution with 1024 input units, 64 filters, kernel widths (5,1) and padding of (3,0).
2. 2D convolution with four filters (matching the number of classes) and a kernel width of one. No padding is needed in this specific case.
3. Conditional random field[20] (crf) for predictions. The forward–backward algorithm was used to calculate each individual and marginal residue label probability of the at the specific position in its protein. The global most likely label

assignment for the entire sequence was done via Viterbi decoding.

SignalP5.0 uses a bidirectional LSTM as well as taxonomic group information to enrich the data of the model. Since data enrichment has already been done by SecVec we skipped this and previous layers/steps. Also different is the usage two dimensional convolutional networks in order to capture and compare different residue properties in one protein (recall that SecVec creates a vector of 1024 for each residue). Therefore, specific padding (3,0) is needed. To avoid overfitting[21], dropout[22], class weighting as well as batch normalization[23] layers have been added. In class weighting, every class have been valued $1/N_{class}$ where N is the amount of Signal peptides per class. Partly, the loss function is calculated through the negative log likelihood between true and predicted labels returned by the forward function of the conditional random field implementation. The calculated parameter is added to the cross entropy between predicted and true observed labels. Further reading on the negative log likelihood and the used programming suite is available on GitHub:
https://github.com/kmkurn/pytorch-crf
All model-parameters were optimized using Adam optimization tool[24].

The previously described techniques improve the performance on both the signal type and the cleavage site prediction though it is noteworthy how the addition of a conditional random field improves the results: Compared to a model without, it enhances both predictions modes on a global (protein) as well as residue level for all organisms. We also observed that the total number of signal peptide gaps and mixed typed predictions are drastically reduced. Gaps in signal peptide predictions and mixtures of the three types are not biologically meaningful and should not happen (ref[7]). A crf uses contextual information from neighboring labels in order to increase the amount of information the model has and to make a good prediction for the next residue. In theory a crf thus should be well suited for this task which coincides with our findings. Still even after application of the crf a few of such problems remain. In both cases simple postprocessing helps: First gaps are filled based on the surrounding signal peptides. Secondly, if a mixed type prediction is present, lower frequent types are replaced by the most abundant one. It is notable that such postprocessing works significantly better on a model without a crf since more unrealistic predictions are made. Table 2 shows a performance comparison between a model using a crf and one without after postprocessing. Hyperparameters of both models are optimized through simple cross validation and grid search where alternating one of five splits have been used to validate different parameterized models. The grid search revealed that mini batches of size 128, a learning rate of 0.01 and 17 learning epochs are fit best for the training of both models. Further information regarding model architecture, parameterization of the models and model comparisons to each other and other algorithms can be found under the results and supplementary information chapters.

Table 2. Comparison between the performance of a model with and without usage of a crf after post processing. Δ CS = mean residue deviation from the actual cleavage site. Both models have been trained with mini batches of size 128, a learning rate of 0.01 and 91 learning epochs. A detailed comparison between crf and non-crf models can be found in the results.

	WITH CRF	**WITHOUT CRF**
Δ CS	0.514 +-0.061	1.149+-0.190
RESIDUE MCC	0.893+-0.006	0.820+-0.007
GLOBAL MCC	0.906+-0.006	0.821+-0.007
RESIDUE (Q4) ACCURACY	97.6%+-0.2	94.8%+-0.2
GLOBAL (Q4) ACCURACY	99.0%+-0.1	98.0%+-0.1

3 Results

Our CNN with and without a conditional random field was trained on one Nvidia GTX 1080 GPU with 8 GB memory. Training, testing and benchmark were split according to data processing described in SignalP5.0. The models reached highest convergance at 17 epochs without overfitting. Figure 2 in the supplementary information shows a loss performance development over 120 epochs of both models and based on this, the early stop has been chosen. Besides early train stopping, the risk of overfitting has been further reduced by using dropout and batch normalization.

The performance scores shown in the following boxplots and confusion matrices are created by using bootstraping[25] and the model with crf as it achieved better scores than the one without. A detailed comparison of the model with and without crf is available under chapter 3.2. Regarding bootstrapping, first, predictions of all benchmarks have been pooled togther. Secondly, we randomly sampled 8809 (amount of predictions) proteins from the population, whereby the same protein can be drawn multiple times. Then the new statistic measurements for mcc, Q4 and confusion matrices are calculated. The second step is repeated 1000 times in order to get a reliable normal distribution for a final estimation of the mean and standard

error of the model results. Finally, mcc scores are compared per organism to other state of the art techniques.

3.1 Model performance

Per-residue performance: SignalP5.0 uses deep learning architectures as well as taxonomic data to predict signal peptide properties. Currently, it is one of the best methods at predicting signal peptides is SignalP5.0 with mcc scores ranging from 0.868 to 0.977 for different peptide types in different organisms. They achieved an overall residue mcc score of 0.94 compared to the 0.893 +- 0.006 (figure 1) which we have been elevated in the course of this work. The accuracy (Q4) of our model hovers around 99.0%+-0.1.

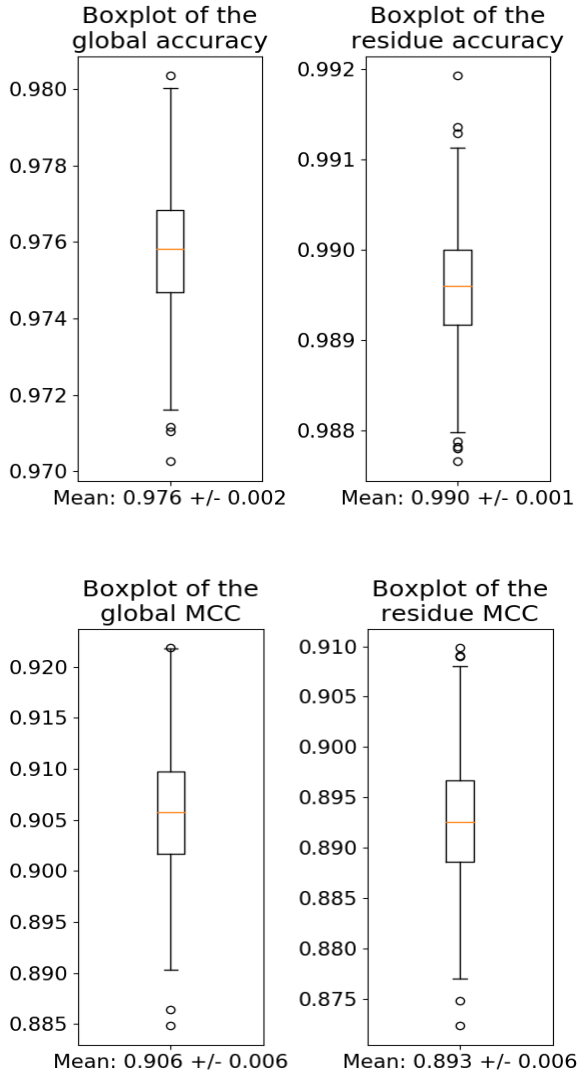

Figure 1 shows a boxplot of the distribution of 1000 mcc and accuracy scores calculated with bootstrapping. The mean mcc lays at 0.906+/-0.006 a global protein level and 0.893+/-0.006 for residue level predictions. The accuracy lays at 97.6 %+/-0.2 and 99.0%+-0.1 respectively.

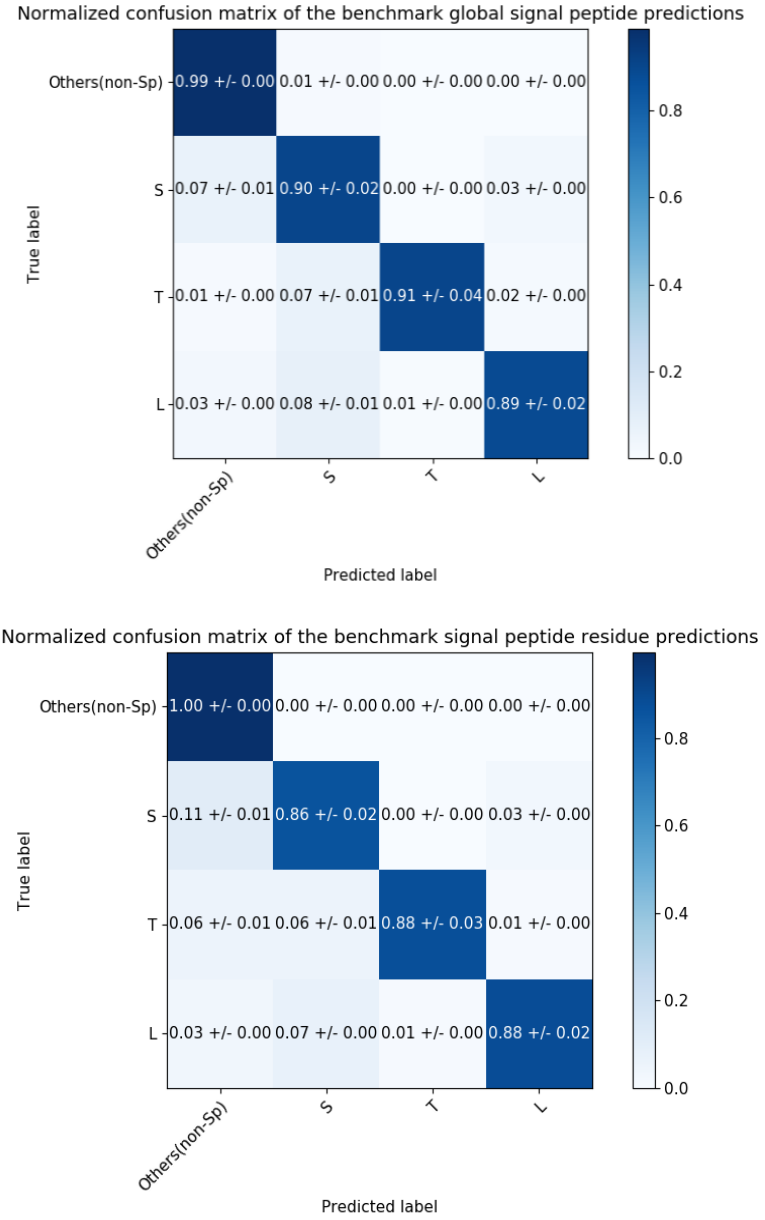

Figure 2 shows the normalized amount of predictions per class on a global protein and residue level with standard error in the form of a confusion matrix. In both cases sp types are predicted at similar rates. Although, it is noticeable that on a global level per-class prediction is improved. Reasons for that are based of inherited class imbalance of the data since it has more influence residue level as there are more labels to be distributed. The model overestimates non signal peptides residues since there are much more common.

Per-protein performance: To gain an estimation of the global, protein level performance the most common signal peptide type per protein is considered for true labels and predictions. Based on these circumstances the statistics are recalculated, whereby the model reaches a mcc of 0.906+-0.006. Q4 is around 97.6%+-0.2. The difference in the global and residue accuracy originates of the class

imbalance of the signal peptide types, as the model overestimates non signal peptides labels. This effect worsens on a residue level because overall more labels must be distributed. Figure 2 emphasis this point since the confusion matrix of the global predictions show overall better estimates of the signal peptide classes than on a residue scale.

For a fair performance comparison, the model has been reduced to most basic architecture that is like the one used in SignalP5.0. Even though SignalP5.0 outperforms our model it has proven that lstms and deep learning architectures like SecVec can capture biophysical properties in embeddings that can efficiently be used for multiple prediction tasks. More complex and adjusted models should be able to achieve on par or even better capacity.

3.2 Performance evaluation of the conditional random field

Since its introduction in 2001[26] conditional random field are combined with lstms to achieve the state of the art results in machine learning including e.g. natural language processing and biological sequences. Here it was used to improve the latter by learning the difference between false and real signal peptide sequences. Examples of false sequences are such with gaps or mixed types which do not occur in vitro.

We used a linear-chain crf implementation taken from GitHub (ref: methods). The implementation uses an input sequence of labels: $y_i = [y_1, \ldots, y_l]$ and predictions $x_i = [x_1, \ldots, x_l]$ where l is the sequence length (in our case 70). The input is fed into the following function:
, where:

$$P(\mathbf{y} \mid \mathbf{X}) = \frac{\exp\left(\sum_{k=1}^{\ell} U(\mathbf{x}_k, y_k) + \sum_{k=1}^{\ell-1} T(y_k, y_{k+1})\right)}{Z(\mathbf{X})}$$

- ***(U)* emissions or unary scores:** represent how likely is y_k given the input x_k,
- ***(T)* transition scores:** represent how likely is that y_k is followed by y_{k+1},
- ***(Z)* a partition function:** normalizes the results in order to get a probability at the end which looks like the following:

$$Z(\mathbf{X}) = \sum_{y'_1} \sum_{y'_2} \cdots \sum_{y'_k} \cdots \sum_{y'_\ell} \exp\left(\sum_{k=1}^{\ell} U(\mathbf{x}_k, y'_k) + \sum_{k=1}^{\ell-1} T(y'_k, y'_{k+1})\right)$$

The final output of the crf is a per residue probability for every signal peptide type, whereby the highest is chosen through viterbi decoding[27] to be the de facto prediction.

When comparing the models without and with a crf (results can be found under the supplementary material), the latter convinces with superior performances. Figure 3 emphasises the area in which crfs shine: The addition of a crf nearly doubles the performance of cleavage site prediction from 1.149+-0.190 to 0.514+-0.061. Cleavage sites are important biological properties to predict, because they have major influence on diverse properties (e.g. activity, here: localisation) of a protein. When observing

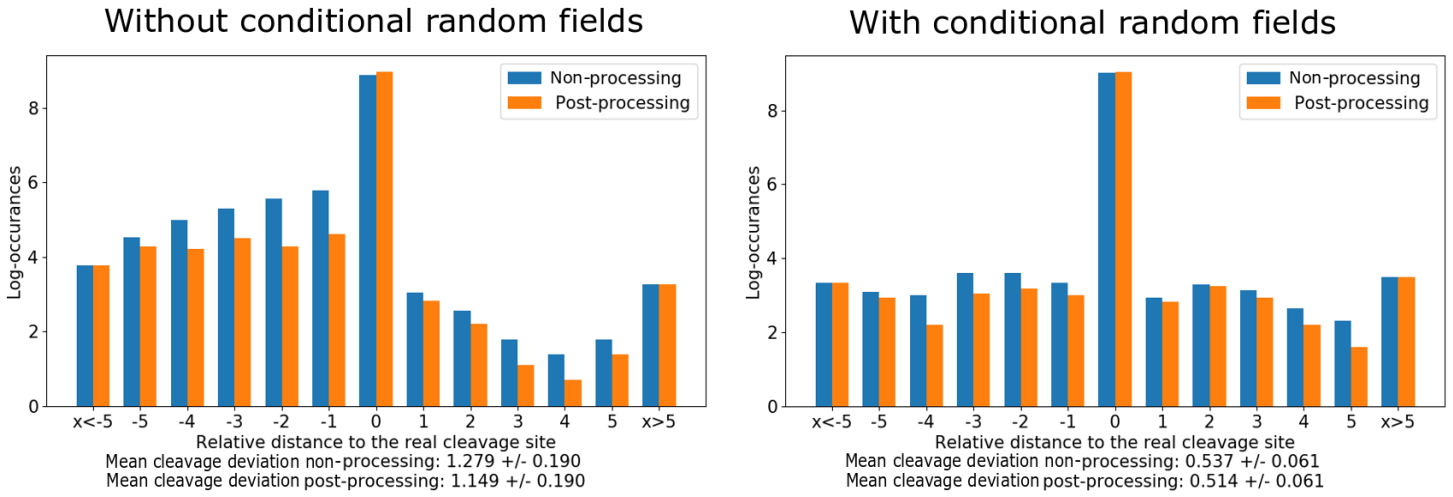

Figure 3 shows the log-occurrences of the relative distances and the mean deviation to the real cleavage site of the signal peptides within bar plots. In both cases most cleavage site predictions are correct (0 distance). Without crfs most false predictions originate of too short or lacking estimations of signal peptides as there are many negative distances. With crf the relative distances are equally distributed with overall more correct predictions. This is also represented in the mean standard deviation of the real cleavage site which is 1.149+-0.190 without and 0.514+-0.061 with crf. It is notable that the post-processing described in the methods section works best for the model without a crf, improving the mean cleavage site predictions by around 0.13. Reasons for that are, that the model creates more implausible sequences that need to be reworked and estimates overall worse predictions. Compared to that crf actively learns correct signal peptide order, which is represented by nearly doubling the performance of predicting the mean cleavage site. Post processing also in not nearly as effective which means that more plausible sequence orders are created.

the efficiency of the post processing it is also noticeable that less predictions need to be polished. The reason for that is, that overall less implausible (gaps, mixed types) but more accurate predictions are made. The model without crf also most likely overestimates non signal peptide residues since most of the time the relative distance to the true cleavage site is negative. With a crf those distances are more equally distributed which is an indicator that the crf also help against overfitting.

On the downside calculating the prediction probabilities and especially the normalization function Z is a computationally complex task ($O(|y|^{\ell})$). Although it can be computed efficiently by dynamical programming (through the so call backward and forward algorithm) the time needed for convergence of one learning epoch is significantly higher (~10 times longer). However, we observed that models with crf also usually reach high convergence at earlier time steps with plateauing performances (e.g. residue mcc of 0.9 after 10 epochs). In the end one must balance between execution time and performance.

3.3 Comparison to SignalP5.0 on an organism level

In figure 4 can be seen that a simple cnn model trained on SecVec embeddings can reach on par or better signal peptide prediction scores in specific organisms compared to four other models. The models are Philius[28], LipoP[29] and SignalP5.0 (ref[2]). The model only gets outperformed by SignalP5.0 at predicting gram-negative bacteria, archaea and eukaryote signal peptide residues. Only predictions of gram-positive bacteria are worse than most of the other models. In this field it is expected that LipoP outperforms because it is specifically tailored on lipoproteins in gram-positive bacteria.

Overall, the results in this manuscript show that we were able to train a model which is a good generalist at predicting signal peptides. Embeddings created through SecVec and more general new machine learning models such as the lstm will significantly influence the field of protein analysis and predictions. Thereby, evolutionary information might soon only play a supporting role. Furthermore, present algorithms that create evolutionary data are unable to handle unlabeled and big data in reasonable time. When a lstm is fully trained it can extract different protein properties for prediction in a fraction of the time. Next, task in the field is now to apply the output to more simple and general machine learning algorithms and possibly support them with additional data to gain state of the art performances.

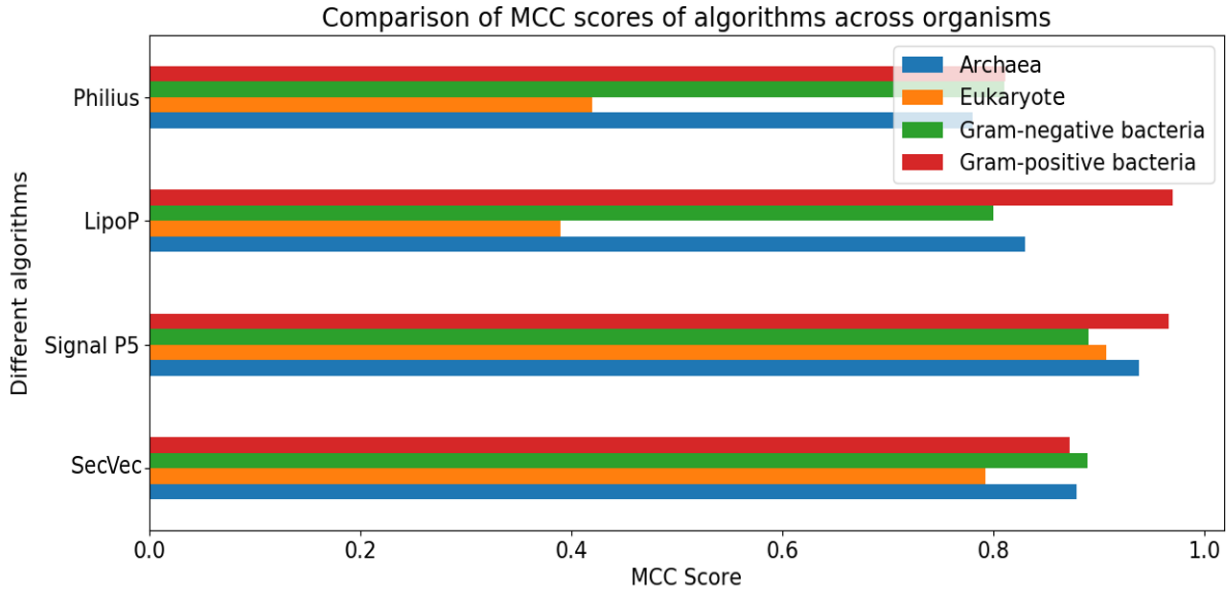

Figure 4 shows a mcc score performance comparison between organisms and prediction methods. Our model with crf build upon SecVec embeddings and data provided by SignalP5.0 reaches on par or better performances to the other models regarding gram-negative bacteria. At predicting archaea and eukaryote signal peptides our model only gets outperformed by SingalP5.0. The worst predictions of the model happen for gram-positive bacteria. Although we do not reach highest scores at any organism, the model is a good generalist at predicting different signal peptides. Scores might be able to be further increased with by feeding the model additional information or using a more advanced and adjusted architecture.

4 Supplementary material

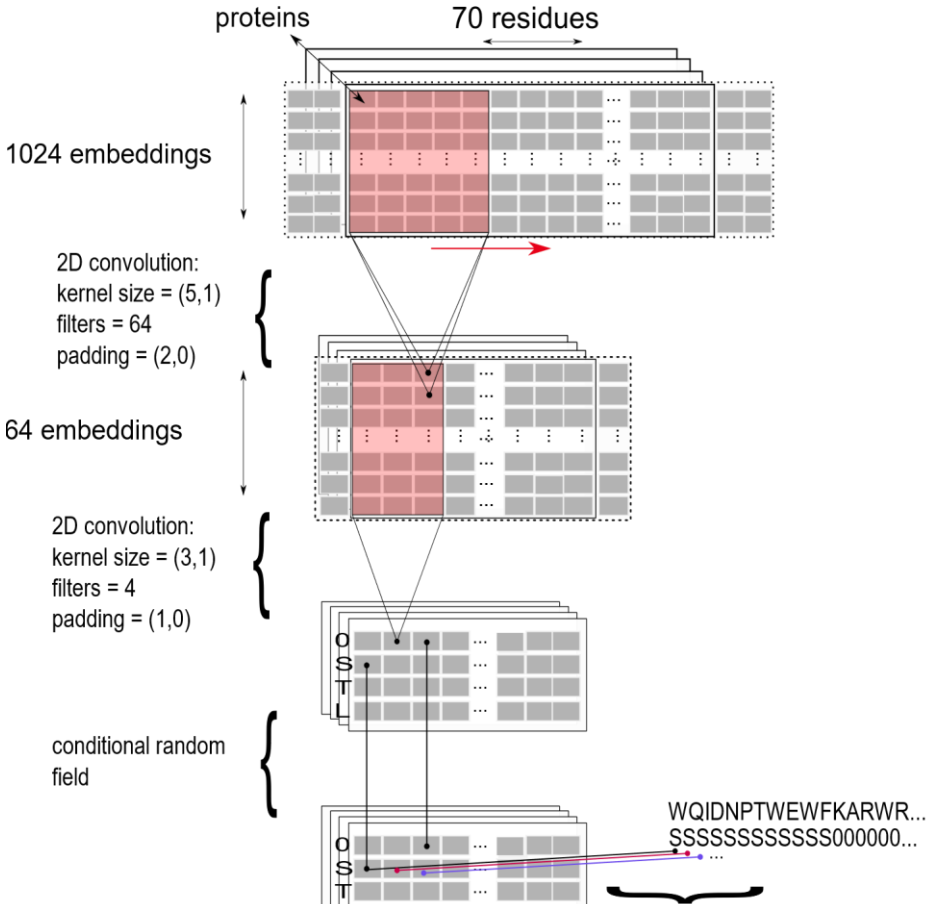

Figure 5 shows the model architecture. The model consists of a 2d convolution that captures signal peptide properties in 64 filters. Those are transformed to per residue predictions by applying a 2d convolution with filters per residue type (non-signal, L, S and T). For both methods according padding is applied. Finally, to find real signal peptide sequences, a conditional random field has been applied that creates final per residue signal peptide predictions via Viterbi decoding.

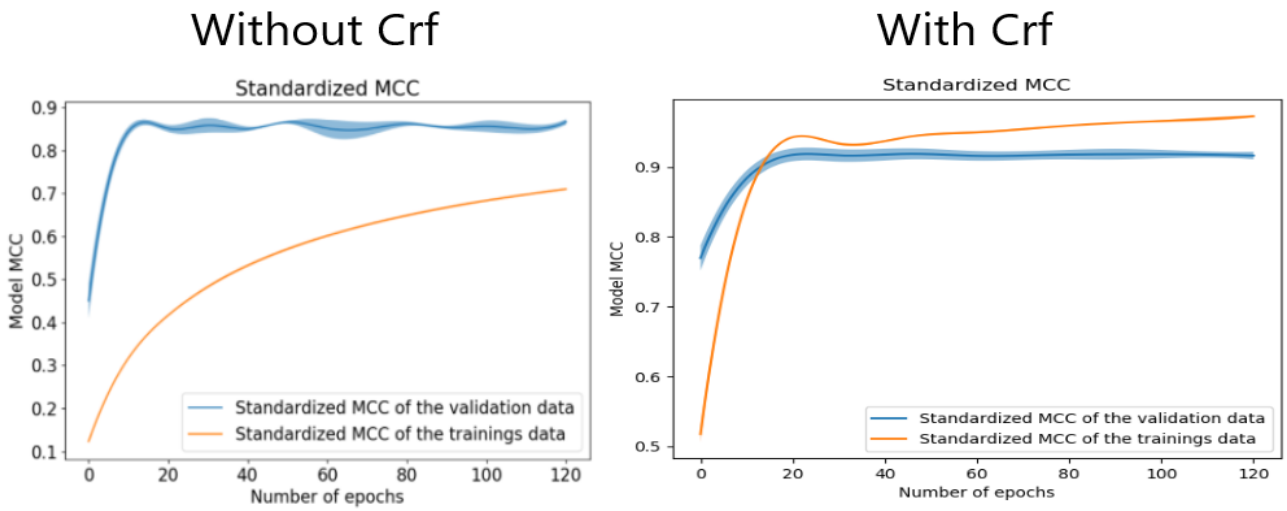

Figure 6 depicts the standardized mcc of the validation data for the models with a crf and without. Around 17 epochs can be seen that the convergence for both models is the highest. Also, for comparative reasons both models have been stopped early in the learning process at around 17 epochs.

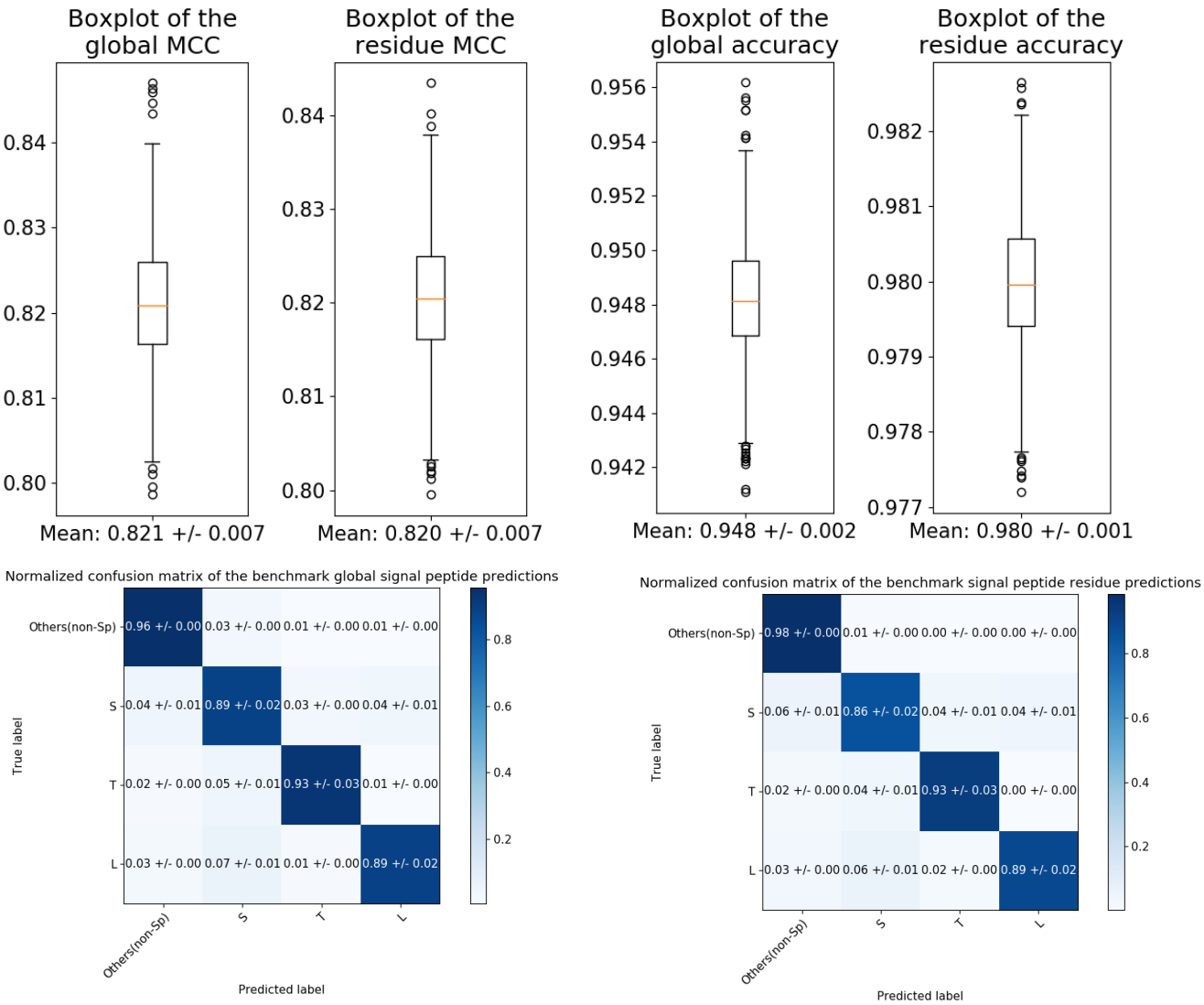

Figure 7 shows the performance scores of a model without a crf. Overall worse performance scores are achieved with a residue mcc of 0.820+-0.007 and a global one of 0.820+-0.007. Respectively, accuracy lays at around 98.0%+-0.1 and 94.8%+-0.2. Confusion matrices reveal that overall worse class predictions are made.

5 References

(1) Heinzinger, M.; Elnaggar, A.; Wang, Y.; Dallago, C.; Nechaev, D.; Matthes, F.; Rost, B. *Modeling the Language of Life - Deep Learning Protein Sequences* 360, 2019.

(2) Almagro Armenteros, J. J.; Tsirigos, K. D.; Sønderby, C. K.; Petersen, T. N.; Winther, O.; Brunak, S.; Heijne, G. von; Nielsen, H. SignalP 5.0 improves signal peptide predictions using deep neural networks. *Nature biotechnology* **2019**, *37*, 420–423.

(3) Weinreb, C.; Riesselman, A. J.; Ingraham, J. B.; Gross, T.; Sander, C.; Marks, D. S. 3D RNA and Functional Interactions from Evolutionary Couplings. *Cell* **2016**, *165*, 963–975.

(4) Henikoff, S.; Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America* **1992**, *89*, 10915–10919.

(5) Nielsen, H.; Engelbrecht, J.; Brunak, S.; Heijne, G. von. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein engineering* **1997**, *10*, 1–6.

(6) Nielsen, H.; Krogh, A. Prediction of signal peptides and signal anchors by a hidden Markov model. *Proceedings. International Conference on Intelligent Systems for Molecular Biology* **1998**, *6*, 122–130.

(7) Laroum, S.; Duval, B.; Tessier, D.; Hao, J.-K. A Genetic Algorithm for Scale-Based Translocon Simulation. In *Pattern Recognition in Bioinformatics*; Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M.,

Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Shibuya, T., Kashima, H., Sese, J., Ahmad, S., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; pp 26–37.
(8) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.
(9) Peters, Matthew E., Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, Zettlemoyer, Luke. *Deep contextualized word representations*, 2018.
(10) Hayat, S.; Sander, C.; Marks, D. S.; Elofsson, A. All-atom 3D structure prediction of transmembrane β-barrel proteins from sequences. *Proceedings of the National Academy of Sciences of the United States of America* **2015**, *112*, 5413–5418.
(11) Marks, D. S.; Hopf, T. A.; Sander, C. Protein structure prediction from sequence variation. *Nature biotechnology* **2012**, *30*, 1072–1080.
(12) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics (Oxford, England)* **2015**, *31*, 926–932.
(13) Steinegger, M.; Meier, M.; Mirdita, M.; Voehringer, H.; Haunsberger, S. J.; Soeding, J. *HH-suite3 for fast remote homology detection and deep protein annotation*, 2019.
(14) Perdigão, N.; Heinrich, J.; Stolte, C.; Sabir, K. S.; Buckley, M. J.; Tabor, B.; Signal, B.; Gloss, B. S.; Hammang, C. J.; Rost, B.; Schafferhans, A.; O'Donoghue, S. I. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences of the United States of America* **2015**, *112*, 15898–15903.
(15) Schafferhans, A.; O'Donoghue, S. I.; Heinzinger, M.; Rost, B. Dark Proteins Important for Cellular Function. *Proteomics* **2018**, *18*, e1800227.
(16) Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* **1975**, *405*, 442–451.
(17) UniProt Consortium, T. UniProt: the universal protein knowledgebase. *Nucleic acids research* **2018**, *46*, 2699.
(18) Savojardo, C.; Martelli, P. L.; Fariselli, P.; Casadio, R. DeepSig: deep learning improves signal peptide detection in proteins. *Bioinformatics (Oxford, England)* **2018**, *34*, 1690–1696.
(19) Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. Automatic differentiation in PyTorch. [Online] **2017**.
(20) Sutton, C. An Introduction to Conditional Random Fields. *FNT in Machine Learning* **2012**, *4*, 267–373.
(21) Tompson, Jonathan, Goroshin, Ross, Jain, Arjun, LeCun, Yann, Bregler, Christopher. *Efficient Object Localization Using Convolutional Networks*, 2014.
(22) Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Online] **2014**.
(23) Ioffe, Sergey, Szegedy, Christian. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015.
(24) Kingma, Diederik P., Ba, Jimmy. *Adam: A Method for Stochastic Optimization*, 2014.
(25) Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Statist.* **1979**, *7*, 1–26.
(26) Brodley, C. E. *Machine learning*. *Proceedings of the eighteenth international conference;* Kaufmann: San Francisco, Calif., 2001.
(27) G. DAVID FORNEY, JR. The Viterbi Algorithm [Online] **1973**.
(28) Reynolds, S. M.; Käll, L.; Riffle, M. E.; Bilmes, J. A.; Noble, W. S. Transmembrane topology and signal peptide prediction using dynamic bayesian networks. *PLoS computational biology* **2008**, *4*, e1000213.
(29) Rahman, O.; Cummings, S. P.; Harrington, D. J.; Sutcliffe, I. C. Methods for the bioinformatic identification of bacterial lipoproteins encoded in the genomes of Gram-positive bacteria. *World J Microbiol Biotechnol* **2008**, *24*, 2377–2382.

Online sources used (specifically at chapter 3.2 for crfs):
https://towardsdatascience.com/implementing-a-linear-chain-conditional-random-field-crf-in-pytorch-16b0b9c4b4ea