Detecting De Novo Plasmodesmata Targeting Signals and Identifying PD Targeting Proteins*

Jiefu Li[1], Jung-Youn Lee[2,3], and Li Liao[1,3,(✉)]

[1] Department of Computer and Information Science, University of Delaware, Newark DE 19716, USA lijiefu@udel.edu
[2] Department of Plant and Soil Sciences, University of Delaware, Newark DE 19716, USA lee@dbi.udel.edu
[3] Delaware Biotechnology Institute, University of Delaware, Newark DE 19716, USA liliao@udel.edu

Abstract. Subcellular localization plays important roles in protein's functioning. In this paper, we developed a hidden Markov model to detect de novo signals in protein sequences that target at a particular cellular location: plasmodesmata. We also developed a support vector machine to classify plasmodesmata located proteins (PDLPs) in Arabidopsis, and devised a decision-tree approach to combine the SVM and HMM for better classification performance. The methods achieved high performance with ROC score 0.99 in cross-validation test on a set of 360 type I transmembrane proteins in Arabidopsis. The predicted PD targeting signals in one PDLP have been experimentally verified.

Keywords: Cellular localization · Support Vector Machines · Hidden Markov Models.

1 Introduction

It is well known that proteins after being synthesized have to be transported to their designated cellular location in order to fulfill the biological functions. However, much detail of the transporting mechanisms remain unknown, and subcellular localization prediction is an active research area in bioinformatics [1, 10, 15].

Plasmodesmata (PD) are membrane-lined intercellular communication channels through which essential nutrients and signaling molecules move between neighboring cells in the plant. This cell-to-cell exchange of molecules through PD is fundamental to the physiology, development and immunity of the plant and is a dynamically regulated cellular process. Several types of endogenous proteins, including type-I transmembrane proteins, as well as numerous PD-targeted proteins derived from plant viruses have been identified to associate with PD. However, no universal or consensus PD-targeting signal has ever been discerned nor molecular details are known as to how integral membrane proteins, including

* The work is funded by National Science Foundation NSF-MCB1820103.

the best characterized PD-located proteins (PDLPs), are targeted to PD-specific membrane domains. As such, the current computational tools for cellular localization prediction do not even have PD categorized as a target location [1, 9].

So far, only eight PDLPs have been experimentally verified in Arabidopsis thaliana, and these proteins share a signature topology, as depicted in Fig. 1. Further experiments (unpublished) have narrowed PD targeting signal(s) down to a region, called extracellular juxtamembrane domain (JMe), which is between the DUF26 extracellular domain and the transmembrane domain (TMD). This region spanned about 20-amino acid residues, 9 AA downstream of the last conserved Cys residue of the DUF26 domain. Our experimental data (unpublished) pinpointed that the JMe region of PDLP5 contained a sufficient primary structure for its targeting to PD. Intriguingly, the data also implicated the presence of a second signal outside of the JMe region. However, the multiple sequence alignments of the eight Arabidopsis PDLP paralogs reveals no hints at the conserved amino acid residues or recognizable patterns shown in Fig. 2.

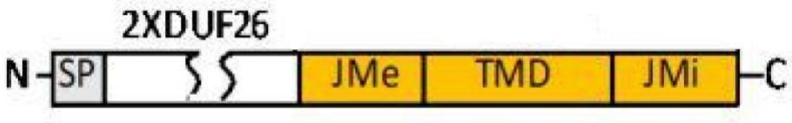

Fig. 1. Structure of PDLP

In this work, we set out to develop computational approaches to: i) detect subtle patterns that are associated with PD targeting, and ii) identify unknown PDLPs in Arabidopsis and other species. The second task can be considered as a classification problem, like other subcellular localization prediction problems. We adopted Support Vector Machine (SVM) [13] as a classifier and with dipeptide features to characterize proteins sequences. The performance of this straightforward approach is surprisingly good.

The first task of detecting PD targeting signal(s) turns out to be more challenging. Using the TMMOD tool [20] with customized training and feature selection, we detected some signals at the vicinity of the last conserved Cys of the DUF in addition to the initially defined JMe region. As for PDLP5, this prediction was consistent with the presence of a second signal. Further phylogenetic analysis of orthologues of PDLPs from other plant species and MEME motif finding [2] also identified similar but slightly stronger patterns in the same location for a subset of PDLPs (consisting of PDLP1 to PDLP4), as shown in Fig. 3.

These findings from the computational analyses together with the preliminary experimental data for PDLP5 prompted us to hypothesize that: there is the second PD-targeting signal outside of the JMe region might reside at the C-terminal end of the DUF domain. Unlike TMD, which has a clear-cut boundary

at both ends from computational predictions, the C-terminal end of the DUF domain is not experimentally defined. This is why the previous experiments limited JMe as 20 AA adjacent to the N-terminal side of TMD, excluding 9 AA to avoid overstepping into the DUF domain. Based on the new hypothesis that this region likely contains a secondary signal for PD targeting, we newly define the JMe as the 30-AA region located between the rightmost conserved Cys at the very C-terminal end of the DUF domain and the N-terminal end of TMD.

Based on the aforementioned hypothesis, we then built a hidden Markov model (HMM) [3–7] to capture the functional structure of the JMe. The model has three states: state α for the left PD signal, state β for the right signal, and state γ for the non-functioning linkers. Using the trained hidden Markov model, we decoded the JMe region of the eight PDLPs, and the following-up experiments have verified the two PD signals and their relative positions in PDLP1, PDLP3, PDLP5 and PDLP8, as predicted by the model. Ongoing experiments are being conducted to verify predicted PD targeting signals for the remaining four PDLP proteins. Furthermore, the model was tested with predicting potential PDLPs in a dataset containing 360 type I transmembrane proteins, and showed remarkable performance as measured as ROC score in cross-validation.

Given the fact that PD targeting signals reside in JMe, we incidentally discovered a pitfall with the SVM classifier, when testing with randomized JMe to establish a baseline. SVM mistakenly classified these synthetic sequences – which are the same of the real PDLPs except for the JMe being randomized. To mitigate this issue, we propose a way to combine SVM and HMM to further improve the classification performance.

The paper is organized as follows. In section 2, we describe in details the structure of the HMM, its training, and integration with SVM. In section 3, we present the results on testing the hidden Markov model and SVM alone and the two methods in tandem. Conclusions are presented in the last section.

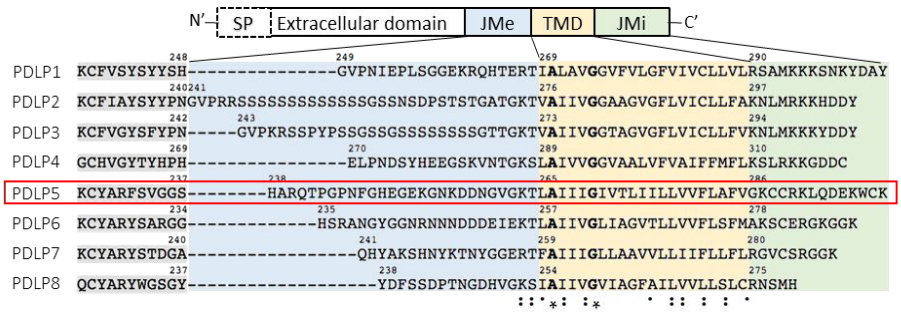

Fig. 2. Alignment of PDLP JMe, TMD, and JMc regions

Fig. 3. MEME motifs discovered for PDLP JMe regions. The red region and green region are α state, β state in our HMM correspondingly.

2 Methods

As mentioned in Introduction, we are tasked with detecting PD targing signals in the sequence of PDLPs and identifying potential novel PDLPs in Arabidopsis and other plant species. In this section, we describe in details the computational methods we developed for these two tasks. Our methods are based on two powerful machine learning methods – SVM and HMM. With the understanding the issues of using either SVM or HMM alone, we describe a way of combinating these two learning algorithms for better classification performance.

2.1 SVM with Dipepetide Features

SVM are a type of classifiers that take vectorized inputs and find the optimal separating hyperplane in the vector space with the positive training examples on one side of the hyperplane and negative training examples on the other side [14]. The optimization is to 1) maximize the margin between the hyperplane to the support vectors, namely, these training examples that are closest to the hyperplane, and 2) minimize the penalty incurred from misclassification. For data that are not linearly separable, the kernel technique can be used, which maps the input to a higher dimension space (called feature space), where the data become linearly separable. Once trained, an unseen data point can be mapped to the input space or the feature space, and based on its relative position to the hyperplane, its classification can be correspondingly made: positive if on the positive example side; negative if otherwise.

SVM have been successfully applied to many classification tasks in bioinformatics, including cellular localization prediction, such as MultiLoc2 [9], SlocX [23], APSLAP [24]. For our task, we adopted SVM with a linear kernel. Since the multiple sequence alignment (see Fig. 2) does not show clear high conservation patterns, we chose to use the alignment-free features to characterize the proteins for classification, specifically the dipeptide features [8]. In our case, the choice of dipeptide feature is a result of balancing the number of features and the number of training examples: dipeptide features capture more information than single amino acid composition and require less examples than tripeptide features and

higher order k-mers, which have a dimension at 8000 and higher and require significantly more training examples.

The dipeptide features are the occurrence frequencies of all 400 possible amino acid pairs in a given protein sequence. Let $x_{dipeptide}$ denote the dipeptide feature, then $x_{dipeptide}(i,j)$ is the frequency of ith and jth amino acid as a neighboring pair to appear in the protein sequence. It is calculated as following:

$$x_{dipeptide}(i,j) = \frac{count([aa_i, aa_j])}{\sum_{i=1}^{20}\sum_{j=1}^{20} count([aa_i, aa_j])} \tag{1}$$

As a common practice, the dipeptide features are normalized as follows:

$$x_{dipeptide-norm} = \frac{x_{dipeptide} - \overline{x_{dipeptide}}}{std(x_{dipeptide})} \tag{2}$$

Although SVM with dipeptide features can be a powerful classifier, as shown in literature [9, 23, 24], it is worth noting that dipeptide occurrence frequency captures global features of the sequence as a whole and is hence not suitable for picking up subtle features from within short regions that are nevertheless important to the protein's functions. And this issue is exacerbated with insufficient amount of positive training data. In such cases, many dipeptide pairs may have zero counts in Eq. 1 – are these zeros real or will they become non zero should enough training examples be available? Consequently, SVM trained on these dipeptide features can be susceptible to overfitting and thus does not generalize well on unseen data.

In our case, we have only eight PDLPs and the JMe region that is known to contain PD-targeting signal is a very short region (around 30 amino acids long) as compared with the whole sequence (up to 700 amino acids long). As such, special attention should be paid to alleviate the aforementioned issues of training SVM with dipeptide features, for otherwise it may give rise to false positive predictions for the sequences that have similar dipeptide features but do not contain PD-targeting signals. Note that these issues are not unique only for SVM but for any classifiers that rely on features from full length sequences. For comparison, we also trained random forest classifier on dipeptide features, and the performances between the two classifiers are comparable, with SVM being slightly better.

2.2 3-States HMM on JMe

In contrast to the dipeptide approach of capturing more global information, the hidden Markov model we designed is focused on JMe region in order to detect PD-targeting signals that the web-lab mutagenesis experiments have suggested.

Based on the signature topology as shown in Figure 1, the JMe region of a PDLP, or a potential PDLP such as type I transmembrane protein, can be easily extracted by two step procedure: 1) finding protein's transmembrane region via various sophisticated tools such as TMDOCK [21] and TMMOD [20], and 2) extracting 30 amino acids upstream of the transmembrane domtain in step 1.

Our hidden Markov model has the 3 states, denoted as α, β, and γ. State α stands for PD-targeting signal A, and state β stands for PD-targeting signal B. State γ stands for everything else in the JMe region but the PD-targeting signals. The hidden states transition connections of these 3 states are shown in Fig. 4. The direct edge between state α and state β is to allow the case in which there are no linker residues between the two PD signals. Since we do not known for any given residue, which of the three states it is – in other words, the training data are unlabeled – we cannot train the HMM with counting as in a typical maximum likelihood approach, or with a multiple sequence alignment as in typical profile hidden Markov models for protein family classification. Instead, we adopted Baum-Welch algorithm, which is an expectation maximization approach and does not require the hidden states to be labeled in the training data [6].

After the model is trained, it is used for two tasks: 1) classifying PDLPs from a set of 360 type I proteins in Arabidopsis thaliana; 2) decoding the JMe region, i.e., marking out each residue as α, β or γ state.

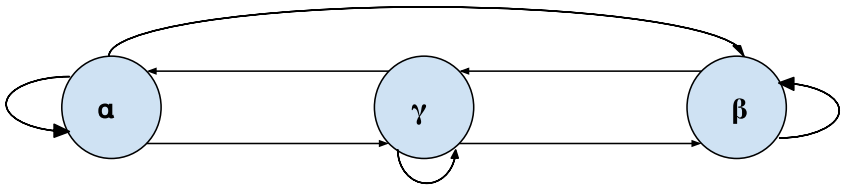

Fig. 4. 3-state HMM hidden states connections

It is worth to note that, because our 3-state HMM focuses on modeling with JMe, the model loses the global picture of PDLP sequences. For example, not every protein sequence with valid JMe region can be considered as PDLP. Furthermore, because HMM is not a discriminative model, the boundary between predicted positive and predicted negative is not readily given by the model and hence can be fuzzy. In the next subsection, we proposed a way to combine SVM and HMM, where we devised a mechanism to impose a threshold on HMM prediction score.

2.3 Combination of SVM with HMM

From the two previous subsections, we can see that SVM is good at capturing characteristic amino acid composition for full length sequences with dipeptide features, but less effective for short sequences like JMe region. This can potentially lead to false positive in predicting some protein sequences as PD targeting even though these proteins do not contain PD-JMe region. The issue becomes more apparent when testing the trained SVM to make de novo prediction of PDLPs in large dataset, in which more proteins may contain dipeptide features or even overall topology similar to that of the real PDLPs. On the other hand,

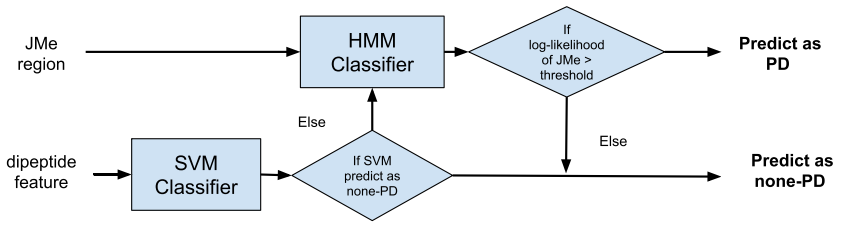

Fig. 5. Pipeline to combine SVM and HMM.

because the HMM focuses on the JMe region exclusively, it loses other features of PDLPs, e.g., it is found experimentally that a mutant without a proper JMi can not even be synthesized.

As such, the shortcomings of SVM and HMM both can give false positive predictions, but for different situations. The SVM tends to give false positive predictions for the sequences with none-PD JMe and PD like dipeptide feature, whereas HMM tends to give false positive predictions for the sequences with PD-JMe but none-PD's other domains because HMM only focuses on JMe region. Therefore, it is sensible to combine SVM and HMM in a complementary way that can overcome their shortcomings with their advantages. An option is to use decision tree on the predictions from SVM and HMM alone to choose a better one. However, due to the limited number of data samples, training decision tree by traditional method did not work. Fortunately, with the understanding of SVM and HMM in this particular task, we proposed a simple decision tree and the decision boundary for each node can be calculated properly, without use of a large training set.

The structure of combining SVM and HMM through decision tree to make final prediction is shown in Fig. 5. The node of SVM's decision boundary is naturally defined by the support vectors in the trained SVM. The decision boundary of HMM node is slightly tricky to define. Since we have the trained HMM model, synthetic JMe sequences can be generated. Given a large number of synthetic JMe sequences, their log-likelihoods follow a Gaussian distribution. By traditional statistic convention, 95% confidence interval is used, and we pick the lower bound of the interval as the decision boundary of HMM node.

3 Results

3.1 Datasets

Dataset A contains 360 type I membrane proteins in Arabidopsis thaliana, including the 8 PDLP sequences, all retrieved from Uniport [12]. From Uniport, labels of protein cellular localizations and their transmembrane domain are extracted, which account for 7 different types of membrane proteins, including endoplasmic reticulum membrane, Endosome membrane, Golgi membrane, Plasma

membrane, Vacuole membrane, Vesicle membrane, and the PD membrane. Table 1 lists each cellular localizations and the number of proteins in that category. In this work, for our purpose, the eight PDLPs are grouped into the other class (PD) and the rest are lumped into one class (none-PD).

In order to train the HMM for the JMe region, a procedure described in the Method section is applied to all sequences in the dataset to extract a valide JMe region defined as 30 amino acids upstream of the transmembrane domain identified by TMMOD. This procedure eliminates Endosome and Golgi membrane proteins from the dataset, as their JMe region is shorter than 30 amino acids.

Table 1. Different types of proteins in dataset A

protein localization	endoplasmic reticulum	Plasma	PDLP	Vacuole	Vesicle
number of proteins	17	322	8	7	6

To test the robustness of SVM and HMM in handling false positives, as described in the Method section, we add to the dataset eight synthetic sequences, which are the 8 PDLP sequences but with the origin JMe region being replaced with random residues. Since JMe region contains PD-targeting signal, it is highly confident that randomly replacing JMe region will lead to none-PD proteins. In other words, these eight synthetic sequences are negative data. Dataset A plus these eight synthetic sequences give rise to dataset B. Note that these eight synthetic sequences will be only used for testing.

As there are only 8 PDLP sequences, the leave-one-out (or equivalently 8-fold) cross validation scheme is adopted to ensure the maximum possible number of training examples to train the models. Specifically, each one of the 8 PDLP sequences is reserved as a positive test example once, and the remaining 7 PDLPs are used as positive training examples. The 352 none-PD sequences, as the negative examples, are randomly split into 8 subsets of equal size (44 sequences). One negative subset is picked to combine with the positive testing example to form the test set (45 sequences); and the remaining 7 negative subsets are merged together to form the negative training set. Note that, for the HMM, no negative training examples are needed. When dataset B is used, the whole process is the same, except that the synthetic none-PD sequences are repeatedly used as testing data for each fold.

3.2 Performance Metrics

To test the trained hidden Markov model $M(\theta)$, the test examples from the 8-fold cross-validation are combined and ranked by their prediction score $P(x|M(\theta))$, which is the likelihood for sequence x to be emitted from the model, calculated from the Forward algorithm. If a threshold is set for the prediction score, test examples with score above the threshold are classified as positive – they are true positive (TP) if their ground truth label is positive; they are false positive

(FP) if otherwise. Similarly, test examples with score below the threshold are classified as negative – they are true negative (TN) if their ground truth label is negative; they are false negative (FN) if otherwise. We use receiver operating characteristic (ROC) curve and ROC score, which is the area under the ROC curve, to evaluate the performance. The ROC curve plots the true positive rate against the false positive rate at the threshold sliding down the ranked list of test examples [18]. ROC curve starts (0,0) and goes to (1,1) in a monotically manner. The perfect classifier has ROC score 1.0, and a random classifier has ROC score 0.5. When a natural choice of threshold is available, like the distance to the the separating hyperplane in SVM, we also use the precision and recall associated with that threshold to evaluate the performance.

3.3 Evaluation: SVM Alone

In this experiment, we train and test a SVM with linear kernel on dipeptide features extracted from full length sequences. The ROC score from 8-fold cross-validation is 1.0 for dataset A but drops to 0.8837 for dataset B, because of misclassifying the synthetic none-PD as PD, which confirms the our concern that SVM alone can be susceptible to the overfitting issue. As comparison, the performance from a RF classifier is: ROC score = 0.9984 for dataset A and ROC score = 0.9177 for dataset B.

3.4 Evaluation: HMM Alone

In this experiment, we train the 3-state HMM as described in the method section. The trained HMM is then tested with 8-fold cross validation the ROC score is 0.93 for dataset A and is 0.9408 for dataset B. Unlike SVM, HMM's performance remains about the same for both datasets, confirming that HMM is more robust with false positives.

For the decoding task, the standard Viterbi algorithm [19] is used to scan the sequence against the model, trained with PDLP5 and ten of its orthologues, to annotate which residues belong to which of the three states, α, β or γ. So far, there is no ground truth available yet to directly evaluate the triple state annotation within JMe, except for PDLP5 and BAK1, the latter of which is experimentally confirmed as non PD targeting, see Fig. 6. It is very encouraging that the experiments for PDLP5 validated the existence of two PD targeting signals and their delineation is consistent with the annotation made by the model. More web-lab experiments are planned to validate model annotation of other PDLP paralogs.

3.5 Evaluation: Combination of SVM and HMM

From the experimental results in subsection 3.3 and 3.4, it is clear that SVM is susceptiable to the pitfall of misclassifying the synthetic none-PD, whereas HMM is not affected. Also, by comparing the ROC score of SVM and HMM in dataset

```
PDLP5 JMe sequence:  YARFSVGGSHARQTPGPNFGHEGEKGNKDDNGVGKT
PDLP5 HMM decoding:  ααααααααααααγγγγγγγγγγββββββββββββββ
BACK1 JMe sequence:  NTKLTPLPASPPPPISPTPPSPAGSNRITGAI
BACK1 HMM decoding:  γγγγγγγγγαγγγγγγγγγγγγγγαγγγγγγγγαγ
```

Fig. 6. HMM decoding results for PDLP5 and BACK1. Red color and green color in the sequence represent state α, state β regions correspondingly. The region without color refers to state γ.

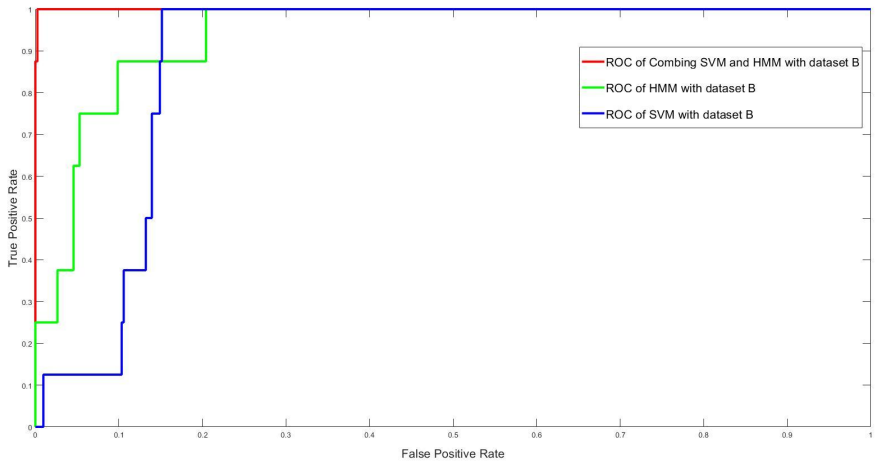

Fig. 7. ROC for combining SVM and HMM with dataset B (red), with ROC score = 0.9997, and comparing with using HMM alone (green), SVM alone (blue)

A, SVM has better performance and it naturally gives a clear decision boundary. In this experiment, we only focus on dataset B to show that by combining SVM with HMM via decision tree, performance can be improved, as compared to either SVM or HMM alone.

Fig. 7 shows the ROC curve for the method of combining SVM with HMM (red) and the comparison with HMM (green) and SVM (blue) alone. The ROC score of combining SVM with HMM is 0.9997. Moreover, the precision/recall and ROC score comparison for all the three method for dataset B are shown in table 2. For comparison, when SVM is replaced with RF, ROC score = 0.9169.

Table 2. Comparison for all method in dataset B

Method	**ROC score**	**precision**	**recall**
SVM with dipeptide feature	0.8837	0.1014	0.8750
HMM with JMe residues	0.9408	not applicable	not applicable
Combining SVM with HMM	0.9997	1.0000	0.8750

4 Conclusion and Future Work

In paper we presented computational approaches based on machine learning techniques to solve a very challenging biological problem: detecting plasmodesmata targeting signals and identifying novel PDLP sequences. The challenges arise from lack of clear sequence patterns, insufficient amount data, and unbalanced dataset. Without addressing these challenges, a straightforward application of standard machine learning techniques can lead to unreliable prediction, as demonstrated with using SVM on dipeptide features. In order to overcome these challenges, we closely incorporated domain specific knowledge into our hidden Markov model design, and devised a pipeline to leverage the predictive power of different models to reduce false positives. As a result, we are able to detect de novo PD targeting signals, verified by wet-lab experiments, and to classify PDLPs with remarkably high accuracy.

It is worth noting that, in this study, we adopted some common practices to avoid overfitting, such as the multi-fold cross-validation scheme and use of a simple linear kernel versus a more powerful kernel in SVM. While the performance from cross-validation as compared with training error does not indicate overfitting, given the small positive training examples in this study, it is difficult to know how well the trained classifiers will generalize to a large data set or data from different genomes, especially in detecting de novo PD proteins, which are actually being investigated in the web-lab experiments and no results to report yet. On the decoding task with our HMM, half of the predicted PD targeting signals have already been verified to be correct in the web-lab experiments, which are ongoing to verify the remaining predicted signals.

An online server will be deployed based the methods in the paper to assist biologists discovering new PDLP members. With new discovered PDLP members, an improved classifier can be built, which leads a positive feedback cycle of PDLP prediction and new PDLP members discovery. For the signal detection task, as the future work, the focus will be finding PD-targeting key residues in JMe region by extracting knowledge from the HMM to help with understanding the PD-targeting mechanism.

References

1. Jose Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017.
2. Timothy L Bailey, Charles Elkan, et al. Fitting a mixture model by expectation maximization to discover motifs in bipolymers. 1994.
3. Leonard Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process. *Inequalities*, 3:1–8, 1972.
4. Leonard E Baum and John Alonzo Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.
5. Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.
6. Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.
7. Leonard E Baum and George Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2):211–227, 1968.
8. Manoj Bhasin and GPS Raghava. Eslpred: Svm-based method for subcellular localization of eukaryotic proteins using dipeptide composition and psi-blast. *Nucleic acids research*, 32(suppl_2):W414–W419, 2004.
9. Torsten Blum, Sebastian Briesemeister, and Oliver Kohlbacher. Multiloc2: integrating phylogeny and gene ontology terms improves subcellular protein localization prediction. *BMC bioinformatics*, 10(1):274, 2009.
10. Kuo-Chen Chou and Hong-Bin Shen. Recent advances in developing web-servers for predicting protein attributes. *Natural Science*, 1(02):63, 2009.
11. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3):462–467, 1968.
12. The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 11 2018.
13. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
14. Nello Cristianini and John Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods*. Cambridge, 2004.
15. Pierre Dönnes and Annette Höglund. Predicting protein subcellular localization: past, present, and future. *Genomics, proteomics & bioinformatics*, 2(4):209–215, 2004.

16. Torben Friedrich, Birgit Pils, Thomas Dandekar, Jörg Schultz, and Tobias Müller. Modelling interaction sites in protein domains with interaction profile hidden markov models. *Bioinformatics*, 22(23):2851–2857, 2006.
17. Alvaro J González and Li Liao. Constrained fisher scores derived from interaction profile hidden markov models improve protein to protein interaction prediction. In *International Conference on Bioinformatics and Computational Biology*, pages 236–247. Springer, 2009.
18. James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.
19. Jerold Heller and Irwin Jacobs. Viterbi decoding for satellite and space communication. *IEEE Transactions on Communication Technology*, 19(5):835–848, 1971.
20. Robel Y Kahsay, Guang Gao, and Li Liao. An improved hidden markov model for transmembrane protein detection and topology prediction and its applications to complete genomes. *Bioinformatics*, 21(9):1853–1858, 2005.
21. Andrei L Lomize and Irina D Pogozheva. Tmdock: an energy-based method for modeling α-helical dimers in membranes. *Journal of molecular biology*, 429(3):390–398, 2017.
22. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman Publishers, San Mateo, CA, 1988.
23. Malgorzata Ryngajllo, Liam Childs, Marc Lohse, Federico M Giorgi, Anja Lude, Joachim Selbig, and Björn Usadel. Slocx: predicting subcellular localization of arabidopsis proteins leveraging gene expression data. *Frontiers in plant science*, 2:43, 2011.
24. Vijayakumar Saravanan and PTV Lakshmi. Apslap: an adaptive boosting technique for predicting subcellular localization of apoptosis protein. *Acta biotheoretica*, 61(4):481–497, 2013.