A MODIFIED STACK DECODER FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Zafer Aydin, Toygar Akgun, Yucel Altunbasak

Center for Signal and Image Processing
Georgia Institute of Technology
Atlanta, Georgia, 30332-0250
Email: {aydinz,takgun,yucel}@ece.gatech.edu

ABSTRACT

Secondary structure prediction is an important step in determining the structure and function of the proteins. A fundamental assumption of the current Bayesian secondary structure prediction methods is the conditional independence of residues which occur in distinct segments [1]. This assumption enables the exact calculation of posterior probabilities by using pre-determined probabilistic models. However, this assumption is clearly violated in the case of protein sequences due to the existence of structural motifs which rely on sequentially distant segments interacting in three-dimensional space, including β-sheets. It has been suggested that the inability to capture such nonlocal interactions may be the main reason for the low accuracy typically achieved in β-strand prediction [1], [2]. Furthermore, the current Bayesian segmentations are based on the maximum a posteriori or marginal posterior mode searches, which return a single segmentation that is optimal in some sense. In this paper, we introduce a new secondary structure prediction method based on a modified version of the well-known stack decoder. The proposed method is an N-best search algorithm which enables us to use the returned multiple segmentations to improve over a single segmentation. Also due to the way the segmentations are constructed it is possible to exploit the non-local interactions between β-strands in a sub-optimal way with the ultimate goal of increasing the overall prediction accuracy.

1. INTRODUCTION

A protein is a biomolecule constructed from amino acid units chosen from a 20 letter alphabet. Protein sequence analysis is an important area where the goal is to predict the structure and function of the newly identified proteins. It has been shown that all the structural information about the protein is embedded in its amino acid sequence. There are several levels at which protein structure prediction can be performed. In secondary structure prediction, one is mainly concerned with the assignment of secondary structure elements to each amino acid residue as shown in Fig. 1. In tertiary structure estimation (i.e., protein folding), the goal is to predict the conformation assumed by protein molecule in 3D space.

The three major secondary structure elements are α-helix $\{\mathrm{H}\}$, β-strand $\{\mathrm{E}\}$ and loop $\{\mathrm{L}\}$. α-helices are strengthened by hydrogen bonds between every fourth amino acid so that the protein backbone adopts a helical configuration. In β-strands the hydrogen bonding is non-local. They adopt a parallel or anti-parallel sheet configuration. Other structural elements such as bends and turns are classified as loops. Therefore a secondary structure prediction assigns for each amino acid a structural state from a 3-letter alphabet {H, E, L}, as depicted in Fig. 1. The secondary structure prediction is an important problem in protein sequence analysis. Accurate predictions provide insights into the molecular structure and function of a protein.

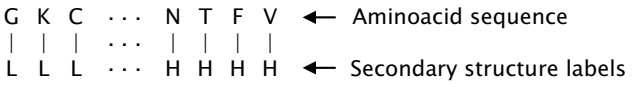

Fig. 1. Aminoacid sequence and the corresponding secondary structure elements.

There are two aspects of secondary structure prediction. In *ab initio* or single sequence prediction, the test sequence does not exhibit significant similarity to any of the training sequences at the sequence level. This is a limiting factor for the prediction accuracy. On the other hand, if there are closely related sequences, this generally implies their structural similarity, and the predictions are improved by considering multiple alignments. In this paper, we propose a new method for the *ab initio* protein secondary structure prediction. Our approach to the secondary structure prediction problem is model-based. We formulate secondary structure prediction in a general Bayesian framework using a semi markov HMM which was introduced in [3] and briefly sum-

This work was supported in part by the Office of Naval Research (ONR) under Award N00014-01-1-0619 and by the National Science Foundation under Award CCR-0113681.

marized here for the sake of completeness.

A fundamental assumption of the current Bayesian secondary structure prediction methods is the conditional independence of residues that occur in distinct segments [1]. This assumption enables the exact calculation of posterior probabilities through the use of pre-determined probabilistic models. However, this assumption is clearly violated in the case of protein sequences due to the existence of structural motifs which rely on sequentially distant segments interacting in three-dimensional space, including β-sheets. It has been suggested that the inability to capture such nonlocal interactions may be the main reason for the low accuracy typically achieved in β-strand prediction. Furthermore, the current Bayesian segmentations are based on the maximum a posteriori or marginal posterior mode searches, which return a single segmentation that is optimal in some sense. In our previous work, we slightly deviated from the independence of amino acid residues that are in distinct segments by incorporating correlations to positions outside the segment [3]. In this paper, we introduce a new segmentation method based on a modified version of the well-known stack decoder. The proposed method is an N-best search algorithm which enables us to use the returned multiple segmentations to improve over a single segmentation. Also due to the way the segmentations are constructed it is possible to exploit the non-local interactions between β-strands in a sub-optimal way with the ultimate goal of increasing the overall prediction accuracy.

2. HMM MODEL

We adopted the semi-Markov HMM introduced in [3]. A secondary structure of a protein can be defined by a vector $(m; \boldsymbol{S}; \boldsymbol{T})$, where m denotes the total number of segments, $\boldsymbol{S}$ represents the segment end positions and $\boldsymbol{T}$ represents the structural state of each segment (α-helix, β-strand or loop). The state prediction could be re-stated as a posterior maximization problem. That is, given the observation sequence of amino acids, denoted by $\boldsymbol{R}$ find the vector $(m; \boldsymbol{S}; \boldsymbol{T})$ with maximum posterior probability $P(m; \boldsymbol{S}; \boldsymbol{T}|\boldsymbol{R})$. Using Bayes rule, this probability could be expressed as follows:

$$P(m; \boldsymbol{S}; \boldsymbol{T}|\boldsymbol{R}) = \frac{P(R|m; \boldsymbol{S}; \boldsymbol{T})P(m; \boldsymbol{S}; \boldsymbol{T})}{P(\boldsymbol{R})} \qquad (1)$$

where $P(\boldsymbol{R}|m; \boldsymbol{S}; \boldsymbol{T})$ denotes the sequence likelihood and P(m; $\boldsymbol{S}$; $\boldsymbol{T}$) represents the a priori distribution. Maximizing the posterior $P(m; \boldsymbol{S}; \boldsymbol{T}|\boldsymbol{R})$ with respect to the state variables is equivalent to maximizing the product $P(\boldsymbol{R}|m; \boldsymbol{S}; \boldsymbol{T})$ $P(m; \boldsymbol{S}; \boldsymbol{T})$. For detailed derivations of $P(m; \boldsymbol{S}; \boldsymbol{T}|\boldsymbol{R})$, $P(R|m; \boldsymbol{S}; \boldsymbol{T})$, and $P(m; \boldsymbol{S}; \boldsymbol{T})$ please refer to [3], and the references therein. For a detailed review on HMMs, see [4]. With the formulation presented in [3], we implemented a semi-Markov HMM. In a typical HMM, there is a finite number of distinct hidden states. Hidden states in our case are structural states {H, E, L}. Starting from an initial state, transitions occur from one state to the other, following a transition probability distribution. At each state an amino acid segment is generated according to the length distribution, and the observation frequency distribution, characteristic to that state. In our implementation of the semi-Markov HMM, we modeled amino acids within the segments according to the correlation patterns discovered by our statistical analysis. We also deviated from the assumption that the individual segments are independent by extending the dependency structure at segment borders and including the correlations to positions outside the segment.

3. STACK DECODER

Stack decoder, introduced by IBM in the 1970s, is a variant of the A^* search [5], a search methodology well-known in the speech recognition society. We can think of stack decoder as a sub-optimal tree search with many appealing properties. The analogy between segmenting a spoken sentence into words and segmenting an amino acid sequence into secondary structure units suggests that the stack decoder can be used in protein secondary structure prediction with the previously discussed semi-Markov HMM model. In speech recognition our observation is the output of some acoustic processor ($\boldsymbol{R}$), basically a string of symbols and our target is the most likely word string in all possible word strings that could have produced the observed speech signal ($\boldsymbol{W}$). In protein secondary structure prediction our observation is the amino acid sequence ($\boldsymbol{R}$) and our target is the most likely secondary structure sequence ($\boldsymbol{W}$). In both cases:

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{R}|\boldsymbol{W})P(\boldsymbol{W}) \qquad (2)$$

As mentioned early, stack decoder is an efficient heuristic for tree searches, and it has been used in speech recognition to find admissible solutions to (2). Although the terms A^* search and stack decoder are used interchangeably, there is a crucial difference between these two, which we will explain when we introduce the original stack decoder algorithm. Before we continue we would like to elaborate on the use of Viterbi decoder in secondary structure prediction. After all, Viterbi search is by far the most popular search method in speech recognition and telecommunications [6]. One of the most appealing properties of the Viterbi search is its low computational complexity, $O(NA^2)$ where N stands for the sequence length to be decoded and A is the alphabet size (no such polynomial bounds exist for A^* search). Furthermore, Viterbi can be made even faster with the help heuristics like beam search. Unfortunately, Viterbi search has its own drawbacks which make us question its usefulness for the protein secondary structure prediction problem.

The Viterbi search is optimal in that the back trace of the highest scoring path obtained by Viterbi search is indeed one of the highest scoring paths. This optimality is guaranteed only if the score of each edge in the path depends only on that edge (due to the fact that Viterbi search is a dynamic programming method). In terms of speech recognition, if we were searching through a word trellis, a bigram language model (score or probability of the current word depends on the previous word) satisfies this requirement because each edge identifies the tow words required to compute the bigram score. However, if we were to use trigram language model (the score or probability of each word depends on the previous two words) we would have to expand the trellis such that each node represents a two-word context in order to use Viterbi and still get optimal results. If we attempt to incorporate longer distance information the number of nodes required to capture the context will increase greatly, which will in turn increase the search complexity greatly, in terms of time and memory. This is the main reason that renders Viterbi search useless, if we are to incorporate non-local interactions. Another serious drawback of Viterbi search is the fact that it yields *the* best scoring answer and *not* a list of best scoring answers which is very desirable in the secondary structure prediction problem. A list of top-scoring answers, also called as an *N-best list*, can be useful if we perform a multi-stage search/segmentation (as proposed in this paper) where the initial stage returns a possibly large number of high scoring candidates and the later stages filters that list with the ultimate goal of obtaining a better result.

The original stack decoder algorithm can be stated as follows:

1. Initialize the stack with a null theory
2. Pop the best (highest scoring) theory off the stack.
3. Perform fast matches to obtain a short list of candidate extensions of the theory.
4. For each extension on the candidate list:
   (a) Perform detailed matches (inter and intra-segment), compute the scores, and determine the highest scoring extension
      i. if (not end-of-sequence) insert into the stack
      ii. if (end-of-sequence) insert into stack with end-of-sentence flag = TRUE.
5. Go to 2.

Here the fast matches in step 3 are computationally inexpensive scoring mechanisms to reduce the number of extensions to be checked with the more computationally expensive detailed matches. The difference between A^* search and stark decoder is the following: In A^* search, we sort the theories in decreasing order of scores, but in stack decoder, we sort the theories first by decreasing order of length, then by decreasing order of scores. Hence, when we drop a theory (using the stack decoder), its the shortest theory with that score.

4. A MODIFIED STACK DECODER

We propose a modified stack decoder which enables us to obtain suboptimal secondary structure segmentations for a given amino acid sequence. Each theory of the stack consists of a secondary structure sequence extended up to position j, where $1 \leq j \leq n$, and n is the total length of the amino acid sequence. We first initialize the stack by including all possible segmentations up to a certain position j^*. Then for each theory, we consider candidate extensions and select a particular extension that maximizes a certain scoring function. We proceed until the n^{th} position is reached, where each theory consists of a secondary structure sequence of length n. Finally, we sort the theories in decreasing order of scores.

Here an extension is obtained by concatenation of a secondary structure symbol (either H, E, or L) instead of a secondary structure segment. This approach allows us to make fair comparisons between the scores of the individual theories. Another advantage of this method is related to the selection of the best extension at a given position. In the case of segment extensions, we are most likely to choose the segments with minimum lengths because for local extensions, the shorter segments have higher probability scores. One way to solve this problem would be to design a score normalization method to compensate for the decrease in the score of a theory due to its length. Unfortunately, such normalization methods are not easy to find and usually hinge on some kind of heuristic, which may not perform good for different protein families. To solve this problem, we are proposing a method that extends the theories by only a single position at each step.

The selection of best scoring extensions from position j to $j+1$ is as follows. We first obtain the list of all possible candidate extensions derived from the entire set of theories[1]. Then we select the particular extension that generates a maximum scoring segmentation terminating by an α-helix segment at position $j+1$. We similarly select for termination by a β-strand and a loop segment at the same position. These three extensions are then carried out in their corresponding theories. After finding the maximum scoring three extensions, we find the next set of three extensions with the second maximum scores and repeat this until all theories are updated. This approach is

[1]Maximum length of the list is 3*N, where N is the total number of theories

similar to Viterbi algorithm, where for each position we find three maximum scoring segmentations (corresponding to three secondary structure states) that terminate at that position. But in our method, we compute not only the maximum scoring segmentations but also the suboptimal ones. This allows us to obtain alternative sequences, which can be used to modify the secondary structure prediction result by fusing information coming from multiple sources.

4.1. How to model non-local interactions due to beta-strands

After obtaining the suboptimal segmentations it is possible to update the scores of each segmentation by including the propensities of amino acid pairs in interacting β-strands (which can be derived from the PDB). After this, a new sort can be performed with the updated scores of each segmentation sequence. We hope that this will improve the prediction accuracy by reordering the segmentations in such a way that the ones with non-local interactions, which have been previously assigned low scores will most likely get higher scores in the successive sorts.

5. RESULTS

We extracted a representative set of 50 proteins from the single-sequence set of the EVA server [2] to serve as the test examples. We used the remaining proteins as the training set to estimate the parameters of the semi-Markov HMM described in [3]. We then computed the Viterbi and the stack decoder segmentations. The Viterbi implementation is similar to the one proposed in by Schmidler *et al.* [1]. The stack decoder implementation selects the best scoring 100 segmentations with suboptimal scores and applies a probabilistic weighting scheme at each particular position. Here the segmentation scores[3] are chosen as the sequence weights and the same score is applied to all positions in a secondary structure sequence. The prediction at a particular position is computed as the secondary structure type with the highest sum of scores.

We compared the sensitivity results of the two algorithms. The sensitivity measure used here is the three-state-per-residue accuracy, (Q_3), which is the total number of correct predictions divided by the total number of amino acids. For the selected set of proteins, we obtained 3% overall improvement in 3-state prediction accuracy (Q_3) in comparison with the Viterbi algorithm.

[2](http://maple.bioc.columbia.edu/eva/doc/ftp.html)

[3]The score of a segmentation is simply the joint probability of observing the amino acid sequence and the secondary structure sequence

	Q_3	Q_α	Q_β	Q_L
Viterbi	58.64	64.79	35.37	69.15
Stack decoder	61.86	65.42	36.78	73.95

Table 1. Sensitivity Results

6. CONCLUSION AND FUTURE WORK

In this work, we implemented a modified stack decoder as an alternative method for protein secondary structure prediction. Using this method, it is possible to obtain suboptimal segmentations of a given amino acid sequence. We showed that the information in suboptimal predictions is useful and can improve the results of the Viterbi algorithm. As a future work, we are going to apply the modified stack decoder algorithm for detecting the non-local interactions in β-strands. Typically protein secondary structure prediction methods suffer from low accuracy in β-strand predictions where non-local correlations have a significant role. With the current method, it is possible to update the score of each segmentation by including the hydrogen bonding propensities of the amino acid pairs in β-strands. We believe that this will compensate the inadequate modeling of β-strand interactions and improve the overall accuracy of the *ab initio* predictions.

7. REFERENCES

[1] S.C. Schmidler, J.S. Liu, and D.L. Brutlag, "Bayesian segmentation of protein secondary structure," *J. Comp. Biol.*, vol. 7, no. 1/2, pp. 233–248, 2000.

[2] D. Frishman and P. Argos, "Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence," *Protein Eng.*, vol. 9, no. 2, pp. 133–142, 1996.

[3] Z. Aydin, Y. Altunbasak, and M. Borodovsky, "Protein secondary structure prediction with semi Markov HMMs," in *2004 IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP'04) Proceedings*, 2004.

[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] D. B. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *1992 IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP'92) Proceedings*, 1992, vol. 1, pp. 25–28.

[6] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Informat. Theory*, vol. IT-13, pp. 260–269, 1967.