Bayesian Protein Secondary Structure Prediction With Near-Optimal Segmentations

Zafer Aydin, *Student Member, IEEE*, Yucel Altunbasak, *Senior Member, IEEE*, and Hakan Erdogan, *Member, IEEE*

***Abstract*—Secondary structure prediction is an invaluable tool in determining the 3-D structure and function of proteins. Typically, protein secondary structure prediction methods suffer from low accuracy in β-strand predictions, where nonlocal interactions play a significant role. There is a considerable need to model such long-range interactions that contribute to the stabilization of a protein molecule. In this paper, we introduce an alternative decoding technique for the hidden semi-Markov model (HSMM) originally employed in the BSPSS algorithm, and further developed in the IPSSP algorithm. The proposed method is based on the N-best paradigm where a set of most likely segmentations is computed. To generate suboptimal segmentations (i.e., alternative prediction sequences), we developed two N-best search algorithms. The first one is an A^* stack decoder algorithm that extends paths (or hypotheses) by one symbol at each iteration. The second algorithm locally keeps the end positions of the highest scoring K previous segments and performs backtracking. Both algorithms employ the hidden semi-Markov model described in Aydin *et al.* [5], and use Viterbi scoring to compute the N-best list. The availability of near-optimal segmentations and the utilization of the Viterbi scoring enable the sequences to be rescored using more complex dependency models that characterize nonlocal interactions in β-sheets. After the score update, one can either keep the segmentations to be employed in 3-D structure prediction or predict the secondary structure by applying a weighted voting procedure to a set of top scoring $M \geq 1$ segmentations. The accuracy measures of the N-best method when used to predict the secondary structure are shown to be comparable or better than the classical Viterbi decoder (MAP estimator), tested under the single-sequence condition. When no rescoring is applied, the stack decoder algorithm with sufficiently large M improves the overall sensitivity measure (Q_3) of the Viterbi algorithm by 1.1%. At the same M value, the N-best Viterbi algorithm improves the Q_3 measure by 0.25% as well as the sensitivity measures specific for each secondary structure type ($Q_{\alpha}^{\mathrm{obs}}$, Q_{β}^{obs}, Q_{L}^{obs}). When the sequences are rescored using the posterior probability distribution computed by the posterior decoding algorithm (MPM estimator), N-best Viterbi improves the Q_3 measure of the Viterbi algorithm by 2.6%. The rescored N-best list approach also enables us to generate suboptimal segmentations that are valid sequences (i.e., realizable from the hidden semi-Markov model). Although the N-best algorithms and the score update procedure brought significant improvements over the Viterbi algorithm, they were not able to outperform the posterior decoding algorithm in the single-sequence condition. Further improvements in the prediction accuracy should be possible with the incorporation of sophisticated models for nonlocal interactions and other physical constraints that stabilize the overall structure of a protein.**

***Index Terms*—Hidden semi-Markov model, N-best list, protein secondary structure prediction, single-sequence prediction, stack decoder, suboptimal segmentations.**

I. Introduction

PROTEIN secondary structure prediction is important as it provides direct insights into the functional role of a protein [6]–[12]. In addition, it can be a step toward the prediction of the 3-D structure [13] or it can be included in fold recognition methods, in which a target amino acid sequence with an unknown structure is compared against a library of structural templates (folds) and the best scoring fold is assumed to be the one adopted by the sequence [14].

The three major secondary structure states are the α-helix $\{H\}$, the β-strand $\{\mathrm{E}\}$, and the loop $\{\mathrm{L}\}$. α-helices are strengthened by hydrogen bonds between every fourth amino acid so that the protein backbone adopts a helical configuration as shown in Fig. 1(a). Likewise in loops (e.g., turns or bends), the hydrogen bonding is mostly local. For example, the turn segment in Fig. 1(b) has a hydrogen bond between the oxygen and hydrogen atoms of the first and the fourth amino acids, respectively. The hydrogen bonding structure in β-strands is slightly different, where both local and nonlocal interactions are observed. In β-strands, the most common local hydrogen bonding is between every two amino acids, and nonlocal interactions are due to hydrogen bonds between amino acid pairs positioned in interacting β-strand segments. A β-sheet is a set of such segments, in which the interacting segment pairs adopt either a parallel or an antiparallel conformation as shown in Fig. 1(c)–(d).

A protein secondary structure prediction algorithm assigns to each amino acid a structural state from a three-letter alphabet $\{H, E, L\}$.[1] There are two types of algorithms in protein secondary structure prediction. A single-sequence algorithm does not use information about other similar (homologous) proteins. The algorithm should be applicable for a sequence with no sequence similarity to any other protein sequence. Algorithms of another type incorporate additional evolutionary information from multiple alignments or multiple alignment profiles, which are derived from homologous proteins [15], [16]. Therefore, the prediction accuracy of such an algorithm should be higher than one of a single-sequence algorithm. The accuracy (sensitivity) of the current state-of-the-art single-sequence prediction methods approaches 70% [5]. The accuracy

[1]There are other alphabets, such as the eight-letter DSSP alphabet (see Section V).

Manuscript received January 26, 2006; revised October 25, 2006. This work was supported by the National Science Foundation–Signal Processing Systems (NSF–SPS) under Grant CCR-0105654. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Elias S. Manolakos.

Z. Aydin and Y. Altunbasak are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: aydinz@ece.gatech.edu; yucel@ece.gatech.edu).

H. Erdogan is with the Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul 34956, Turkey (e-mail: haerdogan@sabanciuniv.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.894404

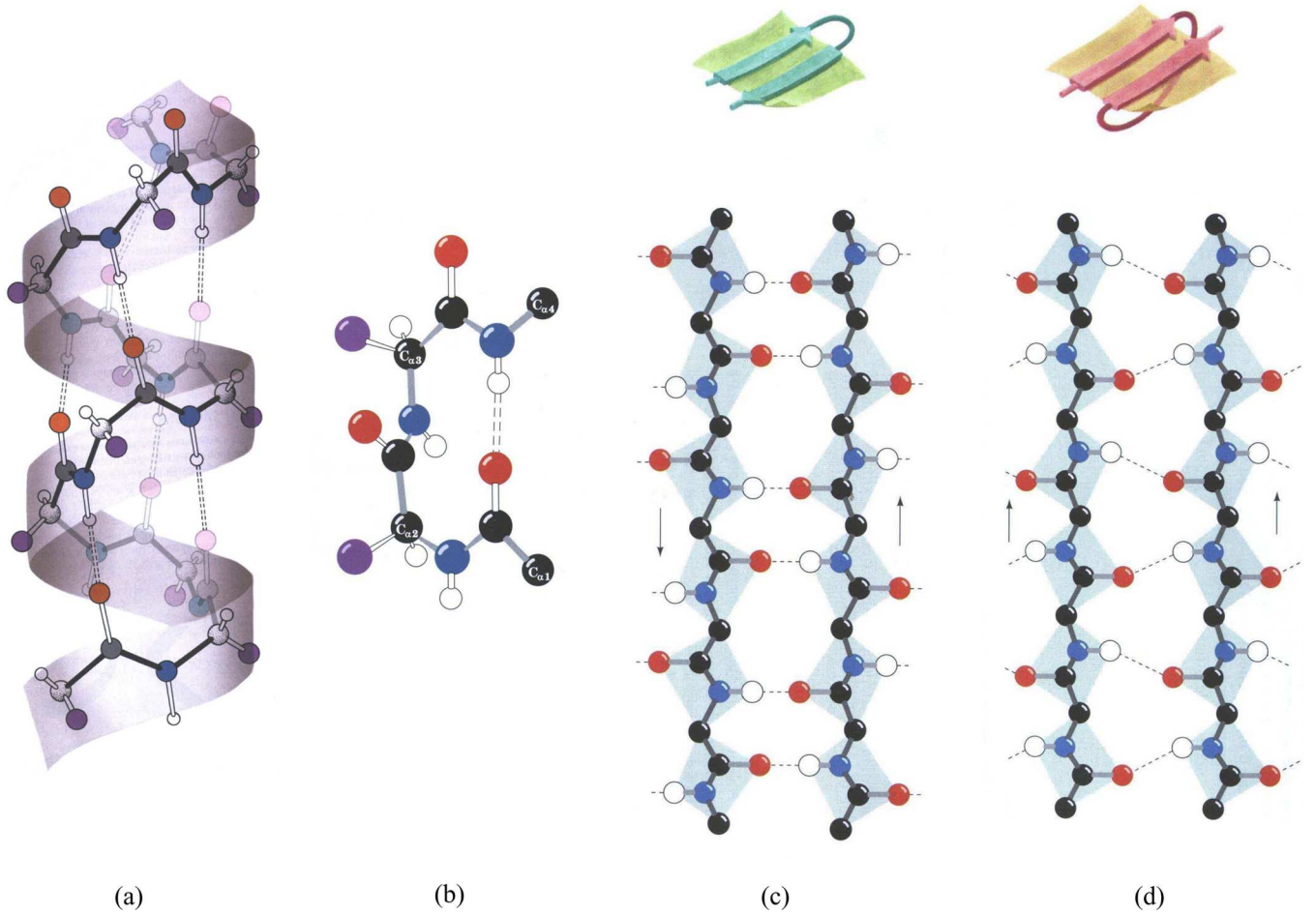

Fig. 1. (a), (b) Local interactions in the α-helix and loop segments. (c), (d) Nonlocal interactions in β-strand segments (the top diagrams illustrate β-strands in cartoon representation). In all diagrams, hydrogen bonds are shown as dashed lines. Solid lines represent covalent bonds. The color representations of the atoms in (a): carbon (C_α) is dark gray, carbon (in $C = O$ group) is light gray, hydrogen is white, oxygen is red, and nitrogen is blue. The color representations of the atoms in (b), (c), and (d): carbon is black, hydrogen is white, oxygen is red, and nitrogen is blue. Side chains are represented as purple spheres. (a) An α-helix segment. (b) A turn segment. (c) Antiparallel conformation. (d) Parallel conformation. (Reprinted from [52]. Illustration, I. Geis. Rights owned by Howard Hughes Medical Institute. Not to be used without permission).

of the state-of-the-art prediction methods that employ multiple alignments or alignment profiles is close to 80% [17]. The secondary structure prediction performance can be further improved by consensus classifiers, in which different prediction methods are combined to improve over a single method [18], [19]. The theoretical limit of the accuracy of secondary structure assignment from an experimentally determined 3-D structure is estimated to be 88% [20]. A real-time analysis and comparison of various protein secondary structure prediction servers can be found at the EVAsec website [21]. A comprehensive evaluation of the protein secondary structure prediction algorithms can be found in Robles *et al.* [18].

Single-sequence algorithms for the protein secondary structure prediction are important because a significant percentage of proteins identified in genome sequencing projects has no detectable sequence similarity to any known protein. Also, many of these hypothetical proteins do not have detectable similarity to any protein at all. Such "orphan" proteins may represent a sizable portion of a proteome.[2] For an orphan protein, any secondary structure prediction method functions as a single-sequence method. Developing better single-sequence prediction methods has a definite merit as it helps us to improve the functional annotation of orphan proteins.

Typically, protein secondary structure prediction methods suffer from low accuracy in β-strand predictions. This is mainly due to the difficulty in modeling nonlocal interactions that are characteristic of β-strands. The β-strand sensitivity of a typical single-sequence prediction method is approximately 25%–50% and that of a method using evolutionary (or homology) information is between 50%–70%. This difference can be explained by the fact that multiple alignments of sequence families or alignment profiles implicitly also contain information about long-range interactions. Further improvements in the prediction accuracy should be achieved by elaborate mathematical models that characterize long-range interactions in β-sheets. Chu *et al.* [2], [3] and Cheng and Baldi [22] combined multiple alignment profiles with nonlocal interaction models. Chu *et al.* [2], [3], extended the work by Schmidler *et al.* [1], [23] and incorporated the multiple alignment sequence profiles into the semi-Markov model. They achieved an overall sensitivity of 72%–74% and a β-strand sensitivity of 56%–59% from a local dependency model with multiple alignment profiles. However,

[2]Proteome is the complete set of proteins that can be expressed by the genetic material of an organism.

they did not report any improvement in secondary structure prediction accuracy through the incorporation of nonlocal interactions. Moreover, their model is based on β-strand segment pair propensities and does not impose global constraints for β-sheet formation. Cheng and Baldi [22] proposed a three-stage modular approach to predict and assemble the β-sheets of a native protein. Their method exploits global covariation and the constraints characteristic of the β-sheet architecture and achieves significant improvements over the existing methods in predicting the β-sheet topology of a protein (i.e., which β-strands are assigned to which β-sheet, conformational ordering of β-strands in each β-sheet and the types of interaction (parallel, antiparallel) between each β-strand segment pair). It assumes that the true secondary structure segmentation is available (either as an experimental sequence or as a prediction) and finds the optimum beta-sheet conformation for that segmentation. However, Cheng and Baldi [22] did not analyze how the derived energy functions can discriminate a false secondary structure from the correct one, and did not apply their method to the secondary structure prediction problem. In single-sequence predictions, Frishman and Argos [4] proposed a method that incorporates a nonlocal interaction model into a nearest neighbor algorithm. Their method achieved an overall accuracy of 68%, which is not significantly higher than the accuracy of the current state-of-the-art methods utilizing local correlations only [5]. Besides, for longer protein sequences with many potential stretches of β-strand residues, the mutual signal from complementary β-strands fades and even the distinction between antiparallel and parallel sheets becomes weak. Therefore, there is still a considerable need to model long-range interactions that contribute to the stabilization of a protein molecule in an attempt to improve the accuracy of the secondary structure prediction.

In this paper, we introduce an alternative decoding technique for the hidden semi-Markov model (HSMM) originally employed in the BSPSS algorithm [1], and further developed in the IPSSP algorithm [5]. The proposed method is based on the N-best paradigm where a set of suboptimal segmentations (N-best list) is computed as an alternative to the most likely segmentation. N-best methods have found diverse applications in speech recognition [24]–[27], sequence-sequence alignments [28]–[30], sequence-structure alignments [31], [32], gene prediction [33], [34], and topology prediction for outer-membrane proteins [35], [36]. To compute suboptimal segmentations, we developed two N-best algorithms: modified stack decoder and N-best Viterbi. The first one is an A^* stack decoder algorithm that extends paths (or hypotheses) by one symbol at each iteration. The second algorithm locally keeps the end positions of the highest scoring K previous segments and performs backtracking. Both algorithms employ a hidden semi-Markov model [5], and use Viterbi scoring to compute the N-best list. The availability of near-optimal segmentations and the utilization of the Viterbi scoring enable the sequences to be rescored using more complex dependency models that characterize nonlocal interactions in β-sheets. After the score update, one can either keep the segmentations to be employed in 3-D structure prediction or predict the secondary structure prediction by applying a weighted voting procedure to a set of top scoring $M \geq 1$ segmentations. The proposed N-best algorithms and techniques can also be applied to other problems that employ hidden Markov models (HMMs), such as video scene annotation, and machine translation.

The organization of this paper is as follows: In Section II, protein secondary structure prediction from the hidden semi-Markov models is described. In Section III, two N-best methods that generate suboptimal segmentations of secondary structure from the hidden semi-Markov model are introduced. Secondary structure prediction by applying rescoring and majority voting procedures is explained in Section IV. In Section V, simulation results on the performances of the N-best algorithms are presented. The extension of the current framework to model nonlocal interactions in β-sheets is discussed in Section VI, followed by the concluding remarks in Section VII.

II. Protein Secondary Structure Prediction With Hidden Semi-Markov Model

A secondary structure of a protein can be defined by a vector $(\mathbf{S}, \mathbf{T})$, where $\mathbf{S}$ is a sequence of the structural segment end positions and $\mathbf{T}$ is a sequence that determines the structural state of each segment (α-helix, β-strand or loop). For instance, for the secondary structure shown in Fig. 2, $\mathbf{S} = (4, 9, 12, 16, 21, 28, 33)$ and $\mathbf{T} = (L, E, L, E, L, H, L)$. Given a statistical model, the problem of protein secondary structure prediction can be stated as the problem of finding the maximum *a posteriori* probability estimator. That is, given an amino acid sequence $\mathbf{R}$,[3] one has to find the vector $(\mathbf{S}, \mathbf{T})$ with maximum *a posteriori* probability $P(\mathbf{S}, \mathbf{T}|\mathbf{R})$, as defined by an appropriate statistical model. Using Bayes' rule, this probability can be expressed as

$$P(\mathbf{S}, \mathbf{T}|\mathbf{R}) = \frac{P(\mathbf{R}|\mathbf{S}, \mathbf{T})P(\mathbf{S}, \mathbf{T})}{P(\mathbf{R})} \tag{1}$$

where $P(\mathbf{R}|\mathbf{S}, \mathbf{T})$ denotes the sequence-likelihood probability and $P(\mathbf{S}, \mathbf{T})$ is the *a priori* probability. Since $P(\mathbf{R})$ is constant with respect to $(\mathbf{S}, \mathbf{T})$, maximizing $P(\mathbf{S}, \mathbf{T}|\mathbf{R})$ is equivalent to maximizing $P(\mathbf{R}|\mathbf{S}, \mathbf{T})P(\mathbf{S}, \mathbf{T})$. Hence, the MAP estimator takes the following form:

$$(\mathbf{S}, \mathbf{T})_{\text{MAP}} = \arg \max_{(\mathbf{S}, \mathbf{T})} P(\mathbf{R}|\mathbf{S}, \mathbf{T})P(\mathbf{S}, \mathbf{T}). \tag{2}$$

To proceed further, we need models for each probabilistic term. We model the *a priori* probability distribution $P(\mathbf{S}, \mathbf{T})$ as follows:

$$P(\mathbf{S}, \mathbf{T}) = \prod_{j=1}^{m} P(T_j|T_{j-1})P(S_j|S_{j-1}, T_j). \tag{3}$$

Here, m denotes the total number of uniform secondary structure segments. $P(T_j|T_{j-1})$ is the probability of transition from a segment with secondary structure type T_{j-1} to a segment with secondary structure type T_j. The third term $P(S_j|S_{(j-1)}, T_j)$

[3] $\mathbf{R} = (R_1, \ldots, R_n)$, where R_i is the ith amino acid.

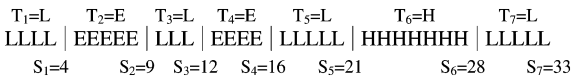

Fig. 2. Secondary structure sequence and its representation by structural segments.

reflects the length distribution of the secondary structure segments by assuming

$$P(S_j|S_{j-1}, T_j) = P(S_j - S_{j-1}|T_j) \tag{4}$$

where $S_j - S_{j-1}$ is equal to the segment length (Fig. 2). The typical form of the segment length distribution for different secondary structure types is illustrated in [1], [3], and [37].

The likelihood term $P(\mathbf{R}|\mathbf{S}, \mathbf{T})$ can be modeled as

$$\begin{aligned} P(\mathbf{R}|\mathbf{S}, \mathbf{T}) &= \prod_{j=1}^{m} P(\mathbf{R}_{[S_{j-1}+1:S_j]}|\mathbf{S}, \mathbf{T}) \\ &= \prod_{j=1}^{m} P(\mathbf{R}_{[S_{j-1}+1:S_j]}|S_{j-1}, S_j, T_j). \end{aligned} \tag{5}$$

Here, $\mathbf{R}_{[p:q]}$ denotes the sequence of amino acid residues with position indices from p to q. The probability of observing a particular amino acid sequence in a segment adopting a particular type of secondary structure is $P(\mathbf{R}_{[S_{j-1}+1:S_j]}|\mathbf{S}, \mathbf{T})$. This term is assumed to be equal to $P(\mathbf{R}_{[S_{j-1}+1:S_j]}|S_j, S_{j-1}, T_j)$. Thus, this probability depends only on the secondary structure type of a given segment, and not of adjacent segments. Note that with this assumption, we ignore the nonlocal interactions observed in β-sheets. On the other hand, this simplification enables us to implement an efficient hidden semi-Markov model.

To elaborate on the segment likelihood term $P(\mathbf{R}_{[S_{j-1}+1:S_j]}|S_j, S_{j-1}, T_j)$, we have to consider the most significant correlation patterns within a secondary structure segment because a fully dependent model is not feasible based on the available training data. To achieve this, we performed a χ^2-test to identify the most significant correlations between amino acid pairs in each type of secondary structure segment. The details of the statistical analysis, and the dimensionality reduction can be found in [5] and [38]. The derived dependency patterns[4] are then used to compute the segment likelihood term as formulated in [1] and [5].

The Bayesian inference approach allows us to implement secondary structure prediction algorithms following the theory of HSMM (see [1], [3], and [5] for details of the HSMM architecture). For instance, the MAP estimation $(\mathbf{S}, \mathbf{T})_{\text{MAP}} = \arg\max_{\mathbf{S},\mathbf{T}} P(\mathbf{S}, \mathbf{T}|\mathbf{R})$ can be found using the Viterbi algorithm. Although a valid state sequence (i.e., realizable from the HSSM), the Viterbi path does not directly optimize the three-state-per residue accuracy (Q_3), which is the commonly used accuracy measure in secondary structure prediction computed as

$$Q_3 = \frac{\text{Total \# of correctly predicted structural states}}{\text{Total \# of observed aminoacids}}. \tag{6}$$

[4]The patterns employed by the semi-Markov methods evaluated in this paper can be found in Supplementary File 1 at http://users.ece.gatech.edu/~aydinz/supp.pdf.

Alternatively, one can determine the sequence of structural states that are most likely to occur in each position, also known as the marginal posterior mode (MPM) estimation. In this approach, the predicted sequence of hidden states is given by $(\mathbf{S}, \mathbf{T})_{MPM} = \arg\max_{(\mathbf{S},\mathbf{T})} \{P(T_{R_i}|R)\}_{i=1}^{n}$, where T_{R_i} is the secondary structure type of the amino acid at position i, and the posterior probability distribution $P(T_{R_i}|R)$ can be computed by the posterior decoding algorithm generalized for an HSMM [1], [39]. Although the prediction sequence obtained by this algorithm might not be a perfectly valid state sequence (i.e., it might not be realizable from the HSMM), the prediction measure defined as the marginal posterior probability distribution correlates very strongly with the Q_3 measure [1]. It has been shown in Aydin *et al.* that the posterior decoding algorithm, when combined with iterative training, can yield sensitivity values $(Q_3(\%))$ around 70.3%, which is one of the best single-sequence results [5]. The performances of the Viterbi and the posterior decoding algorithms are compared in Schmidler *et al.* [1].

III. Suboptimal State Sequences

There are a few methods in the literature that generate an N-best list. These algorithms can be based on an N-best search (e.g., time-synchronous Viterbi-style beam search) [24], [25], on A^* search [26], tree-trellis approach [27], or on divide and conquer methods [40]. Different from the Viterbi algorithm, which finds the most probable state sequence (or path), an N-best method finds the most probable labeling of a given sequence as well as suboptimal labelings (or segmentations). Note that in many applications [25], [33], [36], there can be more than one state sequence that contributes to the same labeling of a given sequence. Therefore, in general, an N-best algorithm always produces a labeling with a probability that is at least as high as the result of the Viterbi algorithm. In the secondary structure prediction, however, there is a one-to-one correspondence between a state sequence and a labeling. In other words, there can be only one state sequence per labeling. Hence, an exact N-best algorithm will produce the Viterbi segmentation as the most likely secondary structure labeling, and the 1-best procedure described in [33] is reduced to the Viterbi algorithm.

In this paper, we developed two approximate N-best algorithms for protein secondary structure prediction that employ hidden semi-Markov models. The first algorithm is a modified stack decoder and the second one is an extension of the Viterbi search. In the next section, we will describe the modified stack decoder algorithm.

A. Modified Stack Decoder

The stack decoder, a search methodology that is well known in the speech recognition literature, was introduced by researchers at IBM [41] and is a variant of the A^* search [26], [42]. One can think of a stack decoder as a suboptimal tree search with many appealing properties. The basic stack decoder algorithm can be found in [26]. The ideas that underpin stack decoding are those of sequential decoding in communications theory [41] and of heuristic search in artificial intelligence [42]. These search algorithms are time asynchronous, in which the

best scoring path or hypothesis, irrespective of time, is chosen as an extension and this process is continued until a complete hypothesis is determined. In the classical implementation of the stack decoder, the stack consists of an ordered heap, which holds a number of partial hypotheses where, in our case, the hypotheses are partial secondary structure sequences. At each iteration, hypotheses of different lengths are extended by one segment and are compared to each other, where only the high scoring ones are kept in the stack as surviving paths.

The crucial function for a stack decoder algorithm is the estimated score (log likelihood) of hypothesis h at time t, and is given by

$$f_h(t) = a_h(t) + b_h^*(t). \tag{7}$$

Here, $a_h(t)$ is the score of the partial hypothesis using information to time t, and $b_h^*(t)$ is the estimate of the best possible score (maximum log likelihood) in extending the partial hypothesis to a valid complete hypothesis. It has been shown that as long as $b_h^*(t)$ is an upperbound on the actual log likelihood, then the search algorithm is admissible [42] (i.e., no errors will be introduced that would not occur if an exhaustive search was performed). This approach allows the hypotheses of different lengths to be compared. However, the disadvantage of approximating $b_h^*(t)$ is the requirement to look ahead at the data. An alternative approach [43]–[45] does not rely on looking ahead. Instead, $b_h^*(t)$ is constructed such that hypotheses with earlier reference times always have higher scores than those with later reference times.

In this paper, we propose a modified stack decoder algorithm to generate suboptimal secondary structure segmentations for a given amino acid sequence. Our approach is similar to the Tailbiting decoder introduced in [46]. In the proposed method, each hypothesis 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. The score of the ith hypothesis with length j is defined as $P\left(\mathbf{R}_{[1:j]}, \mathbf{S}_j^{(i)}, \mathbf{T}_j^{(i)}\right)$, which is the joint probability of observing the amino acid sequence up to position j ($\mathbf{R}_{[1:j]}$), and the secondary structure labeling of the hypothesis $\left(\mathbf{S}_j^{(i)}, \mathbf{T}_j^{(i)}\right)$. Here, $1 \leq i \leq N$, where N is the stack size.

The steps of the algorithm are as follows. We first initialize the stack by including all possible segmentations up to a certain position (j^*) so that the stack contains exactly N segmentations. Then, for each hypothesis, we consider possible candidate extensions and keep the ones with the highest scores. Here, an extension is obtained by concatenating a single secondary structure symbol (either H, E, or L) instead of a secondary structure segment. At each iteration, we extend the hypotheses by one symbol until the nth position is reached, where each hypothesis consists of a secondary structure sequence of length n. Finally, we sort hypotheses in a decreasing order of scores. Stack initialization and hypothesis extension steps of the algorithm are illustrated in Fig. 3. Since the extensions are performed by a single secondary structure symbol instead of a segment, at a given iteration, each hypothesis has the same length. This approach ensures fair comparisons between the scores of the individual hypotheses and eliminates the need to approximate or construct $b_h^*(t)$ in (7). Another advantage of this method is related to the selection of the best extension for a given hypothesis. In the case of segment extensions, we are most likely to choose the segments with minimum lengths because for local extensions, 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 hypothesis due to its length. Unfortunately, such methods usually hinge on some kind of a heuristic, which may not perform well for different protein families. Therefore, we are proposing a method that extends the hypotheses by only a single secondary structure symbol at each iterative step.

The selection of the 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 hypotheses.[5] In computing the extensions, we satisfy the minimum length requirements for the three types of secondary structure. In the current implementation, we restricted the lengths of the α-helices, β-strands and loops to be greater than or equal to 5, 3, and 1, respectively. Before extending a hypothesis, the algorithm checks whether the last segment of its secondary structure sequence satisfies the minimum length requirement. If the length of the last segment is already greater than or equal to the corresponding lowerbound, then all three extensions (H, E, and L) are performed and the extended hypotheses are stored in the candidate extension list. If the last segment is shorter than the lowerbound, then that segment is extended only by its existing secondary structure type and that hypothesis is kept in the stack without being included in the candidate extensions list. If the number of such hypotheses with incomplete extensions is N_s, then the number of hypotheses to be extended for the candidate extensions list is $N_{ct} = N - N_s$ and the total number of hypotheses in candidate extension list N_{ce} becomes $3N_{ct}$. Hence, the set of candidate extensions is derived from those hypotheses in which all secondary structure segments satisfy the minimum length requirements. Having compiled the list of candidate extensions, we compute the score of each hypothesis using the parameters of the hidden semi-Markov model. Finally, we sort the hypotheses in the candidate extension list in decreasing order of scores and insert the first N_{ct} hypotheses back into the stack. Note that for a hypothesis in the candidate extension list, if the extension initiates a new α-helix or β-strand segment, then this extended hypothesis will not satisfy the minimum length requirement. To prevent the score of the new hypothesis to be computed as zero,[6] we modified the length distribution of the α-helices and β-strands for small segments to take nonzero values. We chose a value that is large enough to initiate α-helix and β-strand segments and small enough to avoid paths with dominantly short segments. In the current implementation, the probability of short α-helices $(l_H < 5)$ and short β-strands $(l_E < 3)$ is set to 10^{-5}.

[5]Maximum length of this list is $3 \times N$, where N is the total number of hypotheses or the stack size.

[6]Parameter estimation for hidden semi-Markov model was initially performed using maximum-likelihood estimation procedure on a training set, in which each protein satisfies the minimum length requirements.

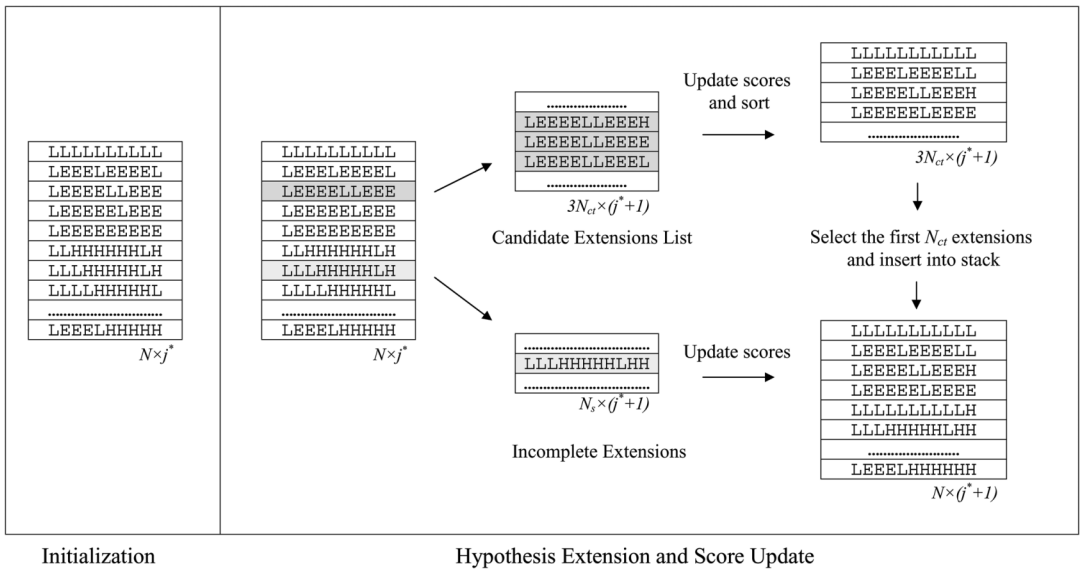

Fig. 3. Modified stack decoder algorithm.

The steps of the algorithm can be summarized as follows.

1) Initialize the stack of size N with all possible extensions up to position $j = j^*$.
2) $j \leftarrow j + 1$.
3) Select a hypothesis and check if the last segment satisfies the minimum length requirement.

a) If the last segment is shorter than the lowerbound, extend the hypothesis with only the type of the last segment, and keep it in the stack.
b) If the last segment satisfies the minimum length requirement, perform all possible extensions (H, E, L) and put the extended hypothesis into the candidate extensions list.
c) Repeat 3 until all N hypotheses are evaluated. If the number of hypotheses that did not satisfy the minimum length requirements is N_s, the number of hypotheses in the candidate extensions list becomes $3(N - N_s)$.

4) Select the highest scoring $N - N_s$ hypotheses from the candidate extension list and insert into stack. Discard the remaining hypotheses.
5) Check if the end of sequence is reached.
a) If (j equal to n), terminate.
b) If (j not equal to n), go to 2.

To evaluate the computational complexity of the algorithm it is useful to divide the operations into two parts: 1) sorting and 2) score computation. To obtain the top scoring $N - N_s$ hypotheses in the candidate extensions list, we use the heap sort algorithm, which has $O(K \log K)$ complexity, where K is the size of the list that is going to be sorted. In C implementation, it takes approximately 30 s to sort a list of 10^6 hypotheses using the heap sort algorithm. Since sorting operations are performed for each position $j = j^* + 1, \ldots, n$, the total number of such operations is $(n - j^*)\overline{N_{ce}} \log \overline{N_{ce}}$, where $\overline{N_{ce}}$ is the average size of the candidate extensions list. In the worst-case scenario, $\overline{N_{ce}}$ takes the value $3N$. Therefore, the computational requirements of the sorting operations are on the order of $O(nN \log N)$. The computational complexity arising from the score computation is on the order of $O(nN)$. For a protein of length 200 amino acids, and a stack of size $N = 30{,}000$, it takes approximately 5 min.[7] to perform all extensions up to the last position and obtain a sorted list.

B. N-Best Viterbi Algorithm

A generalization of the Viterbi algorithm can be used to compute the N-best state sequences. The idea behind the N-best Viterbi algorithm is analogous to the word-dependent N-best algorithm introduced by Schwartz and Austin [24]. In the classical Viterbi algorithm, for each secondary structure segment that is of type $t \in \{H, E, L\}$ and ends at position j, we consider possible previous segments that are of type $l \neq t$ and end at position v. We then store the maximum value of the score function $f(\cdot)$ and the arguments (v, l) where that maximum is achieved. The definition of the score function $f(\cdot)$ is as follows:

$$\begin{aligned} f(v,l,j,t) &= \delta(v,l) P(T = t | T_{\text{prev}} = l) \\ &\quad \times P(S = j | T = t, S_{\text{prev}} = v) \\ &\quad \times P(R_{[v+1:j]} | S_{\text{prev}} = v, S = j, T = t) \\ \delta(j,t) &= \max_{v,l} f(v,l,j,t). \end{aligned} \tag{8}$$

In this equation, $\delta(v, l)$ is the joint probability of observing the amino acid sequence and the secondary structure sequence from position 1 to v. Here, the secondary structure sequence represents the maximum scoring path from position 1 to v, in which the last segment is of type l. The algorithm iterates for positions $j = 1, \ldots, n$, where n is the total number of amino acids in the protein and v can take the values from 1 to $j - 1$.

In the N-best Viterbi algorithm, for each (j, t), instead of storing the maximum value and the arguments of $f(\cdot)$, we rank the possible values of this function with respect to (v, l) and

[7]The computation time is estimated by an Intel Pentium III Processor with a 1.2-GHz CPU.

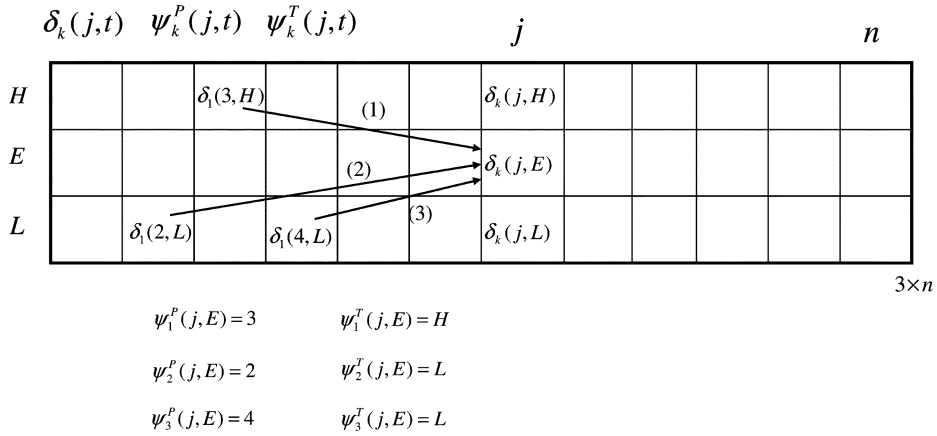

Fig. 4. Forward pass for the N-best Viterbi algorithm.

store the highest scoring K local values as well as the arguments where these values are achieved. Here, K typically takes values from 3 to 6. The difference of this approach from the well known N-best algorithm [25], [33], [36] is that at each state t ending at position j, position indices and types of the K local previous segments are stored instead of the all segment histories (or paths) ending at that position. The recursion for the forward pass can be formulated as follows:

$$\begin{aligned}\delta_k(j,t) &= \mathrm{rank}_k((v,l), f(v,l,j,t)) \\ &= \mathrm{rank}_k((v,l), \delta_1(v,l)P(T=t|T_{\mathrm{prev}}=l)) \\ &\quad \times P(S=j|T=t, S_{\mathrm{prev}}=v) \\ &\quad \times P(R_{[v+1:j]}|S_{\mathrm{prev}}=v, S=j, T=t) \\ \psi_k^P(j,t) &= \arg\mathrm{rank}_k(v, f(v,l,j,t)) \\ \psi_k^T(j,t) &= \arg\mathrm{rank}_k(l, f(v,l,j,t)) \\ k &= 1,\ldots,K. \end{aligned} \tag{9}$$

In this equation, $\mathrm{rank}_k(x,g)$ outputs the kth value of the function $g(\cdot)$ with respect to the argument set x, where $k=1,\ldots,K$. Similarly, $\arg\mathrm{rank}_k(x,g)$ returns the argument set x, where the kth value of $g(\cdot)$ is achieved. $\delta_k(j,t)$ is the joint probability of observing the amino acid sequence and the secondary structure sequence $(\mathbf{S}_j, \mathbf{T}_j)_k$ from position 1 to j. Here, the sequence $(\mathbf{S}_j, \mathbf{T}_j)_k$ does not necessarily correspond to the kth best path from position 1 to j.[8] Instead, it defines a path that satisfies the following constraints: 1) The last secondary structure segment is of type t and ends at position j; 2) the segment before the last segment is of type l_k and ends at position v_k; 3) the segment before the last segment is on the maximum scoring path that ends at v_k with a secondary structure type l_k different from t. The arguments v_k and l_k, where $f(v,l,j,t)$ takes its kth value are stored into $\psi_k^P(j,t)$ and $\psi_k^T(j,t)$, respectively. An iteration of the forward pass is described in Fig. 4. Once the forward pass is completed, we perform backtracking and generate alternative prediction sequences. We start with the nth position and consider all $3K$ segments (K segments for each secondary structure type) that end at this position and are of length $n-v_k$, where v_k is the end position of the previous segment that has been stored in the forward pass. We insert these hypotheses into an array of size N and represent them by character strings, in which the first v_k values are set to "X" and the last $n-v_k$ values are assigned to the secondary structure type of the last segment. Then, for each hypothesis in the array, we consider all possible extensions by one segment in the right-to-left direction, insert the extended sequences into the array, and delete the old sequences. Note that since two adjacent segments cannot be the same in a hidden semi-Markov model, the total number of extensions for each hypothesis is $2K$. If the array becomes full before all of the sequences are extended up to position 1, then we keep those sequences that are already extended completely and extend only the noncomplete sequences. This time, the extensions are performed according to the maximum scoring paths. We terminate when all N sequences are extended up to position 1. The algorithm can be summarized as follows:

1) For each position and secondary structure type, locally keep the end positions of the highest scoring K previous segments.

2) Array initialization: Insert the $3K$ segments that end at position n into the array of size N. Set the array-full flag to FALSE.

3) For each hypothesis in the array with extension-finished flag equal to FALSE:

a) Perform $2K$ back extensions (add segments in the right-to-left direction), in which the previous segment types are different from the type of the current segment.

b) For each back-extension.

i) Check if the total number of hypotheses in the array is N.

A) If the array is full, set array-full flag to TRUE. Do not insert the extended hypothesis into the array. Go to Step 4).

B) If the array is not full and the extension-finished flag is FALSE, insert the extended hypothesis into the array.

ii) Check whether the back-extended hypothesis reaches the first position of the protein. If yes, set the extension-finished flag to TRUE.

[8]The kth path is guaranteed for $k=1$.

C) Repeat step 3 until the array-full flag is TRUE.

4) If array-full flag is TRUE, then repeat Step 3); this time, performing the maximum scoring extensions only until the extension-finished flag is TRUE for each hypothesis.

5) Sort the hypotheses and terminate.

The computational complexity of the algorithm can be evaluated as follows. In the forward pass, for each position j and secondary structure type t that represents a secondary structure segment ending at position j, the highest scoring K local paths are computed. To do this, we need to consider the segmentations such that the end position of the previous segment v takes values from 1 to $j-1$. There are a total of $(n-1)n/2$ such segmentations for $j = 1, \ldots, n$. This requires $3K(n-1)n/2$ operations. Hence, the computational complexity of the forward pass from score computations is $O(Kn^2)$. At each position j and secondary structure type t, we keep two arrays of size K to store the segment end position of the previous segments and the corresponding path scores. To keep K previous segment end positions, a total of $2 \times 3K \times n$ comparisons is required. Hence, the computational requirement to keep K local paths is $O(Kn)$. Backtracking can be performed by a fast recursive procedure. Finally, the sequences are sorted by a heap sort algorithm, with $O(N \log N)$ complexity. For a protein of length 200 amino acids, and a stack of size $N = 30\,000$, it takes approximately 1 min to obtain the sorted list of N suboptimal sequences. The N-best Viterbi algorithm is faster than the modified stack decoder algorithm when additional knowledge sources such as nonlocal interaction models are not utilized. When such dependency models are incorporated, then it might be necessary to update the score of each hypothesis while extending the hypotheses. In that case, the computational complexity of the N-best Viterbi algorithm is expected to increase.

IV. Secondary Structure Prediction Using an N-Best Strategy

The availability of an N-best list enables us to choose from the following options: 1) combine the set of best scoring M segmentations by a weighted majority voting procedure and arrive at a consensus prediction; 2) update the score of each segmentation with more sophisticated functions and compute the final prediction as in 1); and 3) keep the suboptimal segmentations so that they can be used by 3-D structure prediction methods or in expert evaluation. The third option can be considered with or without a score update procedure. In this paper, we propose the utilization of an N-best list to compute the secondary structure prediction for a given amino acid sequence.

To compute suboptimal segmentations, one can use the modified stack decoder or the N-best Viterbi algorithm. Note that, while N-best Viterbi generates the Viterbi result (MAP estimation) as the highest scoring segmentation,[9] the modified stack decoder might not. Therefore, when the modified stack decoder algorithm is used as the N-best list generator, the most likely state sequence is separately computed by the Viterbi algorithm and is included into the N-best list if it scores higher than the top segmentation in the list. On the other hand, when the N-best Viterbi algorithm is used, the MAP segmentation should be contained in the N-best list and there is no need to compute it separately. After the N-best list generation step, the segmentations can be rescored using additional knowledge sources. Here, we investigated two possible scenarios: no rescoring, and rescoring with posterior probability distribution $(P(T_{R_i}|R))$ of the IPSSP method [5], which is computed using the posterior decoding algorithm. In the latter case, the score of a segmentation is computed as the sum of the posterior probability values of the secondary structure states at each position. This can be formulated as follows. Let the jth segmentation be denoted as $\left(\mathbf{S}^{(j)}, \mathbf{T}^{(j)}\right) = \left(T_{R_1}^{(j)}, T_{R_2}^{(j)}, \ldots, T_{R_n}^{(j)}\right)$, where $1 \leq j \leq N$, and $T_{R_i}^{(j)}$ is the secondary structure type of the amino acid at position i. Then, the score of that segmentation is updated as $\mathcal{U}^{(j)} = \sum_i P\left(T_{R_i}^{(j)}|R\right)$.

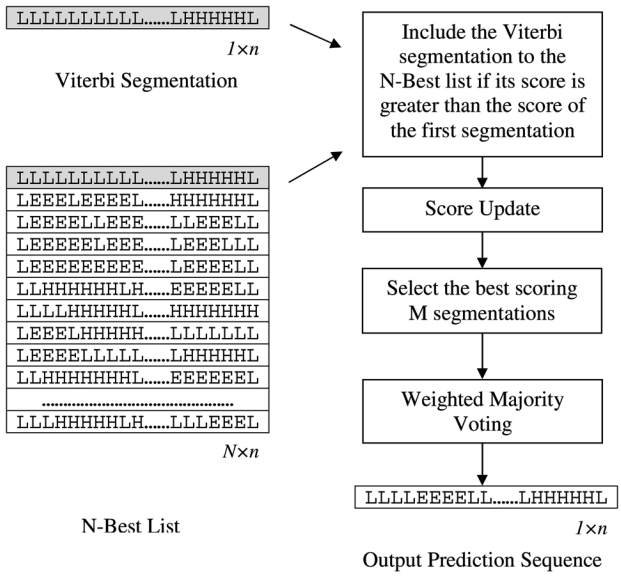

Fig. 5. Secondary structure prediction with near-optimal segmentations. The N-best Viterbi algorithm does not require the extra computation of the Viterbi (MAP) segmentation and proceeds with the score update after the N-best list generation step.

After the score update procedure, the final prediction sequence can be computed by applying a weighted voting procedure to a set of best scoring M segmentations. Here, each sequence is weighted by its segmentation score and the same score is applied to all positions within the sequence. The predicted state at position i $(i = 1, \ldots, n)$ is computed as the secondary structure type with the highest sum of scores. Setting $M = 1$ reduces to selecting the most likely segmentation as the prediction sequence. The steps of the method are illustrated in Fig. 5.

V. Results

In our simulations, we used two datasets. The first one is the set of "sequence-unique" proteins derived from the PDB database, which is available from the EVA server's ftp site: http://cubic.bioc.columbia.edu/eva/doc/ftp.html. This set contained 3324 proteins, as of September 2004, and the proteins

[9]The score of a segmentation is defined as the joint probability of the amino acid sequence and the secondary structure sequence (i.e., $P(\mathbf{R}, \mathbf{S}, \mathbf{T})$.

in this set were selected to satisfy the condition that the percentage of identity between any pair of sequences should not exceed the length-dependent threshold S (for instance, for sequences longer than 450 amino acids, $S = 19.5\%$) [47]. The second set is CB513, which contains 513 nonhomologous protein chains [48]. A copy of the datasets can be found at http://users.ece.gatech.edu/~aydinz/Nbest.html.

In all simulations, we removed sequences that contained secondary structure segments longer than $D = 40$ amino acids because for longer segments, the maximum likelihood estimation for length distribution becomes less reliable due to the small sample size. In simulations with the N-best algorithms, to further refine the datasets and limit the computations, we removed proteins shorter than 30 and longer than 400 amino acids. After applying these constraints, 2251 proteins remained in the EVA set, and 447 proteins in the CB513 set. The eight state secondary structure assignments for the proteins in the datasets were taken from the PDB database.[10] To reduce the eight secondary structure state assignment used in the DSSP notation to three, we used the following conversion rule: H to H; E to E; and all other states to L, which is also known as the "CK" mapping [16], [48]. We also considered using the length adjustments proposed by Frishman and Argos [4] that convert the α-helices shorter than five amino acids and β-strands shorter than three amino acids to loops.

In all simulations, we performed a leave-one-out cross-validation, which is a k-fold cross-validation experiment with k being equal to the number of proteins in the dataset. At each step, a protein is chosen as the test example and is taken out from the dataset. The remaining proteins form the training set and are used to estimate the parameters of the hidden semi-Markov model (i.e., transition, length, and emission distributions). Since the true secondary structures were available, we used the maximum-likelihood estimation procedure where we count the observed frequencies for the desired quantities, and apply a proper normalization factor to compute the probability values. After estimating the model parameters, we predicted the secondary structure sequence of the test protein and repeated the leave-one-out procedure until all proteins are evaluated. Then, we compute the performance measures by taking the true secondary structures of the proteins as reference. To evaluate the performance, we chose the three-state-per-residue accuracy (Q_3) as the overall sensitivity measure, which is computed as the total number of correctly predicted amino acids in all dataset proteins divided by the total number of amino acids in the dataset. The sensitivity measure can also be used for each type of secondary structure. For instance, Q_α^{obs} is computed as the total number of amino acids correctly predicted as α-helix divided by the total number of amino acids observed in α-helix segments [i.e., TP/(TP+FN)]. In addition to the (Q_3) measure, we also used the Segment OVerlap score (SOV), which is based on the average overlap between the observed and the predicted segments instead of the average per-residue accuracy. The SOV measure provides more elaborate scoring in which the predictions that have high per-residue accuracy but deviate from experimental segment length distributions are assigned lower scores (see [49], and [50] for the definition of the SOV measure).

[10]PDB uses the DSSP algorithm for the assignment of the secondary structure from the atomic coordinates.

TABLE I
SENSITIVITY RESULTS OF THE VITERBI, MODIFIED STACK DECODER AND N-BEST VITERBI ALGORITHMS EVALUATED ON THE EVA SET. IN SIMULATIONS WITH THE N-BEST ALGORITHMS, WEIGHTED MAJORITY VOTING IS APPLIED TO A SET OF TOP SCORING M SEGMENTATIONS

Sensitivity	$Q_3(\%)$	$Q_\alpha^{obs}(\%)$	$Q_\beta^{obs}(\%)$	$Q_L^{obs}(\%)$
Viterbi	64.17	64.99	28.70	76.56
Mod. Stack Dec.	65.28	60.42	26.78	79.95
N-Best Viterbi	64.42	65.41	28.74	76.77

A. N-Best Predictors

In this section, we compare the performances of the various methods including the Viterbi algorithm, the N-best method, and the IPSSP method. In the first set of simulations, we did not apply any score update to the N-best list. Then, we evaluated the effect of updating the segmentation scores with the posterior probability distribution $P(T_{R_i}|R)$ obtained by the IPSSP method [5] (at http://users.ece.gatech.edu/~aydinz/supp.pdf). The dependency patterns (feature sets) employed by the evaluated methods can be found in Supplementary File 1. To initialize the frequency tables, Laplace's rule is used as the pseudocount method, in which the entries are set to 1.

1) Modified Stack Decoder Versus N-Best Viterbi: We first compare the performances of the Viterbi, modified stack decoder, and the N-best Viterbi algorithms. The size of the N-best list is chosen as $N = 30\,000$. For simplicity, no score update is applied. To obtain the final prediction sequence, the best scoring $M = 5000$ segmentations are combined by the weighted voting procedure as explained in Section IV. K is chosen as 3 for the N-best Viterbi algorithm. From Table I, the modified stack decoder algorithm performs better than the Viterbi algorithm by 1.1% in terms of the Q_3 measure. For the N-best Viterbi algorithm, the improvement is only 0.25% because the N-best Viterbi generates a significantly higher number of sequences with scores close to the most likely sequence. In addition, the score differences are smaller for the N-best Viterbi algorithm. The overall accuracy of the N-best Viterbi can be improved by increasing the size of the N-best list. A comparison of the structure-type-specific measures shows that the N-best Viterbi algorithm has the highest Q_α^{obs} and Q_β^{obs} values followed by the Viterbi algorithm. The highest loop sensitivity Q_L^{obs} is achieved by the modified stack decoder algorithm. These results show that the information in suboptimal segmentations is useful and is capable of improving over the MAP segmentation even when there is no score update.

2) N-Best List Size: At this stage, we found it useful to investigate the effect of changing the N-best list size N, and the number of voting sequences M. Table II shows the sensitivity results of the proposed method for different values of N and M. Here, suboptimal segmentations are obtained using the N-best Viterbi algorithm with $K = 3$. From Table II, increasing the size of the N-best list improved the Q_3, Q_α^{obs}, and Q_L^{obs} measures. For the same value of N, increasing the number of voting sequences improved only the Q_L^{obs} measure. The results demonstrate that suboptimal segmentations contain valuable information and can improve the accuracy when the segmentations are sampled more densely and the list is deeper. The decrease in the β-strand sensitivity for increasing values of N can be explained

TABLE II
SENSITIVITY RESULTS OF THE N-BEST VITERBI ALGORITHM EVALUATED ON THE EVA SET FOR CHANGING VALUES OF N AND M

N	M	$Q_3(\%)$	$Q_\alpha^{obs}(\%)$	$Q_\beta^{obs}(\%)$	$Q_L^{obs}(\%)$
30,000	500	64.422	65.413	28.750	76.775
30,000	5,000	64.421	65.411	28.748	76.775
50,000	5,000	64.446	65.514	28.559	76.833
100,000	10,000	64.519	65.526	28.544	76.974

TABLE III
SENSITIVITY RESULTS OF THE VITERBI, IPSSP, AND N-BEST VITERBI WITH SCORE UPDATE, EVALUATED ON THE REDUCED EVA SET BY LEAVE-ONE-OUT CROSS-VALIDATION

Sensitivity	$Q_3(\%)$	$Q_\alpha^{obs}(\%)$	$Q_\beta^{obs}(\%)$	$Q_L^{obs}(\%)$
Viterbi	63.95	65.66	24.37	77.30
IPSSP	70.06	66.77	45.25	80.93
N-Best Viterbi Score Update (M = 1)	66.52	65.43	30.83	80.08
N-Best Viterbi Score Update (M = 10,000)	65.80	65.67	29.06	79.18

by the fact that the current statistical model can only capture local interactions, which are dominantly observed in α-helices and loops. Therefore, without incorporating additional knowledge sources, N-best methods will not improve the accuracy of the β-strand predictions.

3) Score Update With Marginal Posterior Distribution: In this section, we investigate the effect of updating the segmentation scores by using the posterior probability distribution $P(T_{R_i}|R)$ as described in Section IV. We compare the performances of the three methods: Viterbi algorithm, IPSSP method, and the N-best method with score update. To compute suboptimal segmentations, we used the N-best Viterbi algorithm with $N = 1\,000\,000$. For the number of voting sequences used in the weighted majority voting step, we chose two different values $M = 1$ and $M = 10\,000$. The number of local suboptimal segments K is set to 4. The results of the cross-validation experiments are shown in Tables III and IV for the EVA set and in Tables V and VI for the CB513 set. In EVA set simulations, we used the original IPSSP method [5], which takes the ensemble average of three dependency models each calibrated by an iterative training procedure. In simulations with the refined CB513 set, we used the IPSSP-simp method, which employs reduced versions of the IPSSP's dependency models (see Supplementary File 1). In both versions of the IPSSP, the threshold used in the iterative training step is set to 35%. For all methods, the Laplacian pseudocount method is applied to initialize the frequency tables. The $P(T_{R_i}|R)$ values that are used to obtain the IPSSP predictions and to update segmentation scores are computed as in Aydin *et al.* [5], in which the posterior probability distributions from three dependency models are averaged (see Supplementary File 1 for the dependency models).

The score update procedure yields an average improvement of 2.6% over the Viterbi algorithm in terms of the three-state-per-residue accuracy $Q_3(\%)$ (Tables III and IV), and an average improvement of 3.5% in terms of the SOV measure (Tables V and VI). In both simulations, choosing the most likely segmentation as the prediction sequence performed better than the consensus approach (weighted voting) on a set of most likely segmentations. This result can be explained by the fact that, after applying a score update, the first M segmentations get more diverse for increasing values of M and, hence, less accurate ones are likely to be selected. Hence, when a score update is performed, it is better to choose the most likely segmentation as the final prediction sequence.

TABLE IV
SOV MEASURES OF THE VITERBI, IPSSP, AND N-BEST VITERBI WITH SCORE UPDATE, EVALUATED ON THE REDUCED EVA SET BY LEAVE-ONE-OUT CROSS-VALIDATION

SOV Score	$SOV_3(\%)$	$SOV_\alpha(\%)$	$SOV_\beta(\%)$	$SOV_L(\%)$
Viterbi	51.45	64.47	27.63	52.57
IPSSP	64.09	70.99	54.63	63.55
N-Best Viterbi Score Update (M = 1)	54.74	66.32	35.48	55.06
N-Best Viterbi Score Update (M = 10,000)	53.55	65.93	33.54	53.67

TABLE V
SENSITIVITY RESULTS OF THE VITERBI, IPSSP-SIMP, AND N-BEST VITERBI WITH SCORE UPDATE, EVALUATED ON THE REDUCED CB513 SET BY LEAVE-ONE-OUT CROSS-VALIDATION

Sensitivity	$Q_3(\%)$	$Q_\alpha^{obs}(\%)$	$Q_\beta^{obs}(\%)$	$Q_L^{obs}(\%)$
Viterbi	61.93	69.44	28.04	73.05
IPSSP-simp	67.91	68.38	48.80	76.65
N-Best Viterbi Score Update (M = 1)	64.69	69.17	36.67	75.01
N-Best Viterbi Score Update (M = 10,000)	63.71	70.10	33.88	73.64

TABLE VI
SOV MEASURES OF THE VITERBI, IPSSP-SIMP, AND N-BEST VITERBI WITH SCORE UPDATE, EVALUATED ON THE REDUCED CB513 SET BY LEAVE-ONE-OUT CROSS-VALIDATION

SOV Score	$SOV_3(\%)$	$SOV_\alpha(\%)$	$SOV_\beta(\%)$	$SOV_L(\%)$
Viterbi	52.34	67.50	31.52	52.27
IPSSP-simp	64.39	70.78	56.44	63.98
N-Best Viterbi Score Update (M = 1)	56.00	67.91	41.44	55.09
N-Best Viterbi Score Update (M = 10,000)	54.47	67.72	38.20	53.51

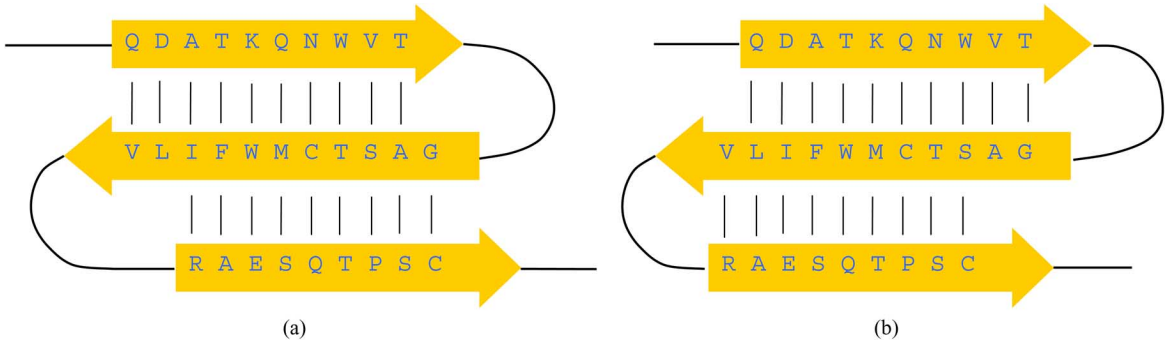

Fig. 6. Two possible patterns for the amino acid pairing of a β-sheet. β-strands are shown as colored segments. The letters correspond to the amino acid residues.

We have demonstrated that when the sequences are rescored with more elaborate functions, it is possible to improve the accuracy of the Viterbi algorithm. Though the N-best method with the score update using the marginal posterior distribution did not perform better than the posterior decoding algorithm, utilization of the N-best approach and the Viterbi scoring has some advantages. First, suboptimal segmentations generated by the Viterbi scoring will be valid sequences (i.e., they will be realizable from the hidden semi-Markov model). Therefore, after applying the score update procedure, it will be possible to obtain segmentations that are both valid and are more accurate than the Viterbi segmentation. In addition, since the correct secondary structure segmentation is also realizable from the hidden semi-Markov model, it can be captured when the size of the N-best list is sufficiently large. As a third and more important factor, the Viterbi algorithm optimizes the joint probability of the amino acid and secondary structure sequences $P(\mathbf{R},\mathbf{S},\mathbf{T}) = P(\mathbf{S},\mathbf{T})P(\mathbf{R}|\mathbf{S},\mathbf{T})$. This property allows one to remodel the *a priori* distribution $P(\mathbf{S},\mathbf{T})$ and the sequence likelihood distribution $P(\mathbf{R}|\mathbf{S},\mathbf{T})$ by incorporating the nonlocal hydrogen bonding propensities in β-sheets and other constraints that define the overall structure of a protein. We expect that when nonlocal correlation models are available, the N-best methods will significantly improve the β-strand predictions, and will contribute to the overall prediction accuracy.

VI. Implications for Modeling Nonlocal Interactions in β-Sheets

We conclude by discussing a possible extension of this framework to model nonlocal interactions in protein structures, providing a possible direction for future improvements in secondary structure prediction accuracy. The proposed Bayesian formulation and the hidden semi-Markov model for protein secondary structure prediction has some limitations due to the assumptions made in the model derivation. For instance, it is assumed that the segment likelihood terms are independent from each other as formulated in (5). This assumption enables us to implement efficient hidden Markov models. However, with this assumption and others inherent in the theory of hidden Markov models, it is not possible to model long-range interactions, especially the nonlocal hydrogen bonds in β-sheets that have a significant role in the stabilization of a protein's structure. More complex dependency models are not feasible due to limitations in the available training data and high computational requirements. To overcome such difficulties, one can follow a two-stage approach. The first step generates a list of best scoring prediction sequences (N-best list) that contains the most likely prediction sequence (MAP solution) as well as those that are suboptimal under a predefined statistical model. Such a model contains local correlation information and is relatively simple. In the second step, the score of each sequence is updated using a nonlocal correlation model, which is an extended version of the initial model and utilizes information related to long-range interactions. The final prediction sequence can then be computed using a weighted voting scheme applied to a selected set of top scoring sequences.

A nonlocal interaction model has to capture the intrinsic properties of β-sheet formation. A β-sheet is a group of β-strand segments, where each group contains at least two β-strands that interact pairwise through nonlocal hydrogen bonds. Within each β-sheet, β-strand pairs can have either parallel or antiparallel interaction as shown in Fig. 1(c)–(d). However, for a given suboptimal segmentation $(\mathbf{S},\mathbf{T})$ that contains at least two β-strands, β-sheet groups, and interaction types are not defined. Therefore, there can be numerous ways to group β-strands into β-sheets, order them spatially, and specify the type of interaction between each segment pair. Moreover, due to the possible length differences between β-strand segments, there can be many alternatives to align amino acid pairs that make hydrogen bonding contacts. In Fig. 6, two possibilities are shown for the amino acid pairing pattern of a β-sheet that has three β-strand segments. To include these constraints into the model, we will modify the computation of the sequence score term $P(\mathbf{R},\mathbf{S},\mathbf{T})$ as follows. Let $(\mathbf{S},\mathbf{T})$ contain r β-strand segments $(\mathcal{B}_1,\ldots,\mathcal{B}_r)$, where $r \geq 2$ and let $\mathcal{C}$ denote the 3-D conformation of these segments that defines the grouping of β-strands into β-sheets, spatial ordering of β-strands, the interaction type of each β-strand segment pair, and the amino acid pairing pattern. Then the score of a segmentation $(\mathbf{S},\mathbf{T})$ can be updated as

$$P(\mathbf{R},\mathbf{S},\mathbf{T}) = \sum_{\mathcal{C}} P(\mathbf{R},\mathbf{S},\mathbf{T},\mathcal{C}) = \sum_{\mathcal{C}} P(\mathbf{R},\mathbf{S},\mathbf{T}|\mathcal{C})P(\mathcal{C}). \tag{10}$$

Using Bayes' rule

$$P(\mathbf{R},\mathbf{S},\mathbf{T}|\mathcal{C}) = P(\mathbf{R}|\mathbf{S},\mathbf{T},\mathcal{C})P(\mathbf{S},\mathbf{T}|\mathcal{C}). \quad (11)$$

In the above equations, $P(\mathcal{C})$ is the *a priori* distribution of a 3-D conformation, $P(\mathbf{S},\mathbf{T}|\mathcal{C})$ is the secondary structure label probability given conformation, and $P(\mathbf{R}|\mathbf{S},\mathbf{T},\mathcal{C})$ is the sequence likelihood term for a given conformation. Note that these terms are quite similar to the ones in the local-dependency model, except for $P(\mathbf{S},\mathbf{T})$ being replaced with $P(\mathbf{S},\mathbf{T}|\mathcal{C})$ and $P(\mathbf{R}|\mathbf{S},\mathbf{T})$ is replaced with $P(\mathbf{R}|\mathbf{S},\mathbf{T},\mathcal{C})$. To model the terms in (11), it is necessary to incorporate the constraints that define the secondary structure including the nonlocal forces in β-sheets. With this motivation, one can update the computation of the sequence likelihood term as follows:

$$P(\mathbf{R}|\mathbf{S},\mathbf{T},\mathcal{C}) = \prod_{T_j\in\{H,L\}} P(\mathbf{R}_{[S_{j-1}+1:S_j]}|\mathbf{S},\mathbf{T}) \times \prod_{k=1}^{w} P(\mathbf{R}_{\mathcal{B}_1},\ldots,\mathbf{R}_{\mathcal{B}_{n_k}}|\mathbf{S},\mathbf{T},\mathcal{C}) \quad (12)$$

where w is the total number of β-sheets in $\mathcal{C}$, such that each sheet contains n_k β-strand segments, and $\sum_k n_k = r$. In the above formulation, the segment likelihoods of α-helices and loops are computed the same as before [see (5)], but those of β-strands are obtained from a nonlocal model. The computation of the joint probability term for a β-sheet can be simplified as

$$P(\mathbf{R}_{\mathcal{B}_1},\ldots,\mathbf{R}_{\mathcal{B}_{n_k}}|\mathbf{S},\mathbf{T},\mathcal{C}) = P(\mathbf{R}_{\mathcal{B}_1}|\mathbf{S},\mathbf{T},\mathcal{C})\prod_{l=2}^{n_k} P(\mathbf{R}_{\mathcal{B}_l}|\mathbf{R}_{\mathcal{B}_{l-1}},\mathbf{S},\mathbf{T},\mathcal{C}). \quad (13)$$

Here, we assume that a β-strand only depends on a neighboring β-strand. This assumption is quite reasonable since in most β-sheets, β-strand segments interact pairwise and form a ladder topology as in Fig. 6. To elaborate further, we should model the terms $P(\mathbf{R}_{\mathcal{B}_l}|\mathbf{R}_{\mathcal{B}_{l-1}}\mathbf{S},\mathbf{T},\mathcal{C})$, $P(\mathcal{C})$, and $P(\mathbf{S},\mathbf{T}|\mathcal{C})$ by including the hydrogen bonding propensities of β-strands and other constraints that stabilize the overall structure of a protein. At this point, we leave the derivation of a complete model and estimation of its parameters as a future work. Before concluding this section, it is worthwhile to note that as the number of β-strands increases, the total number of possible conformations rises exponentially. Therefore, efficient algorithms have to be developed to search the conformation space and update the score of a segmentation. Cheng and Baldi [22] introduced graph-matching algorithms to predict the β-sheet architecture of a given protein (i.e., the β-strand grouping, pairing, and interaction types). These algorithms can be further developed by eliminating the architectures that never occur in real proteins (see Ruczinski *et al.* [51]). The observation frequency of the remaining architectures can be modeled by the $P(\mathcal{C})$ term. As an alternative to the graph-matching algorithms, one can select a representative set of conformations using Monte Carlo sampling similar to the method proposed by Chu *et al.* [2], [3], especially for longer proteins with many potential β-strand residues.

VII. Conclusion

In this work, we developed two N-Best search algorithms for the protein secondary structure prediction though the proposed techniques can be also applied to other problems that employ HMMs, such as gene prediction, topology prediction for outer-membrane proteins, sequence-sequence and sequence-structure alignments, speech recognition, video scene annotation, and machine translation. We showed that the information in suboptimal segmentations is useful and can improve the sensitivity of the Viterbi algorithm (Q_3) up to 1% without applying any score update. When the segmentations are rescored using the marginal posterior probability distribution, the improvement becomes 2.6%. Unfortunately, the two N-best algorithms and the score update procedure were not able to perform better than the posterior decoding algorithm in single-sequence predictions. As a future work, we are planning to develop nonlocal interaction models and incorporate them into the N-best method. Such models will be able to characterize the hydrogen bonding propensities within β-sheets and can further be extended to include other constraints such as solvent accessibility. We expect that this will compensate the inadequate modeling of long-range interactions and improve the overall prediction accuracy.

References

[1] S. C. Schmidler, J. S. Liu, and D. L. Brutlag, "Bayesian segmentation of protein secondary structure," *J. Comp. Biol.*, vol. 7, pp. 233–248, 2000.

[2] W. Chu, Z. Ghahramani, and D. L. Wild, "A graphical model for protein secondary structure prediction," in *Proc. Int. Conf. Machine Learning*, 2004, pp. 161–168.

[3] W. Chu, Z. Ghahramani, A. Podtelezhnikov, and D. L. Wild, "Bayesian segmental models with multiple sequence alignment profiles for protein secondary structure and contact map prediction," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 3, no. 2, pp. 98–113, Apr.-Jun. 2006.

[4] 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.

[5] Z. Aydin, Y. Altunbasak, and M. Borodovsky, "Protein secondary structure prediction for a single sequence using hidden semi-Markov models," *BMC Bioinform.*, vol. 7, no. 178, 2006.

[6] A. Wallqvist, Y. Fukunushi, L. R. Murphy, A. Fadel, and R. M. Levy, "Iterative sequence/secondary structure search for protein homologs: Comparison with amino acid sequence alignments and application to fold recognition in genome databases," *Bioinform.*, vol. 16, no. 11, pp. 988–1002, 2000.

[7] Y. An and A. Friesner, "A novel fold recognition method using composite predicted secondary structure," *Proteins: Structure Function Genom.*, vol. 48, pp. 352–366, 2002.

[8] V. D. Francesco, P. J. Munson, and J. Garnier, "FORESST: Fold recognition from secondary structure prediction of proteins," *Bioinform.*, vol. 15, no. 2, pp. 131–140, 1998.

[9] E. Bindewald, A. Cestaro, J. Hesser, M. Heiler, and S. C. E. Tosatto, "MANIFOLD: Protein fold recognition based on secondary structure, sequence similarity and enzyme classification," *Protein Eng.*, vol. 16, no. 11, pp. 785–789, 2003.

[10] Y. Hou, W. Hsu, M. L. Lee, and C. Bystroff, "Remote homology detection using local sequence-structure correlations," *Proteins: Structure, Function Bioinform.*, vol. 57, pp. 518–530, 2004.

[11] P. Fontana, E. Bindewald, S. Toppo, R. Velasco, G. Valle, and S. C. E. Tosatto, "The SSEA server for protein secondary structure alignment," *Bioinform.*, vol. 21, no. 3, pp. 393–395, 2005.

[12] K. Wang and R. Samudrala, "FSSA: A novel method for identifying functional signatures from structural alignments," *Bioinform.*, vol. 21, no. 13, pp. 2969–2977, 2005.
[13] C. Bystroff, V. Thorsson, and D. Baker, "HMMSTR: A hidden markov model for local sequence structure correlations in proteins," *J. Mol. Biol.*, vol. 301, pp. 173–190, 2000.
[14] L. A. Kelley, R. M. MacCallum, and M. J. E. Sternberg, "Enhanced genome annotation using structural profiles in the program 3d-pssm," *J. Mol. Biol.*, vol. 299, pp. 501–522, 2000.
[15] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *J. Mol. Biol.*, vol. 232, no. 2, pp. 584–599, 1993.
[16] D. Frishman and P. Argos, "Seventy-five percent accuracy in protein secondary structure prediction," *Proteins*, vol. 27, pp. 329–335, 1997.
[17] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri, "Exploiting the past and the future in protein secondary structure prediction," *Bioinform.*, vol. 15, pp. 937–946, 1999.
[18] V. Robles, P. Larrañaga, J. Peña, E. Menasalvas, M. Pérez, and V. Herves, "Bayesian networks as consensed voting system in the construction of a multi-classifier for protein secondary structure prediction," *Artif. Intell. Med. (Special Issue in Data Mining in Genomics and Proteomics)*, vol. 31, pp. 117–136, 2004.
[19] Y. Guermeur, G. Pollastri, A. Elisseeff, D. Zelus, H. Paugam-Moisy, and P. Baldi, "Combining protein secondary structure prediction models with ensemble methods of optimal complexity," *Neurocomput.*, vol. 56, pp. 305–327, 2003.
[20] B. Rost, "Rising accuracy of protein secondary structure prediction," in *Protein Structure Determination, Analysis, and Modeling for Drug Discovery*, D. Chasman, Ed. New York: Marcel Dekker, 2003.
[21] EVA: Secondary Structure (intro) [Online]. Available: http://cubic.bioc.columbia.edu/eva/doc/intro_sec.html.
[22] J. Cheng and P. Baldi, "Three-stage prediction of protein β -sheets by neural networks, alignments and graph algorithms," *Bioinform.*, vol. 21, pp. i75–i84, 2005.
[23] S. C. Schmidler, J. S. Liu, and D. L. Brutlag, "Bayesian protein structure prediction," *Case Studies Bayesian Stat.*, vol. 5, pp. 363–378, 2001.
[24] R. Schwartz and S. Austin, "A comparison of several approximate algorithms for finding multiple (N-Best) sentence hypothesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 1991, vol. 1, pp. 701–704.
[25] R. Schwartz and Y. L. Chow, "The N-Best algorithm: An efficient and exact procedure for finding the N most likely sentence hypothesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 1990, vol. 1, pp. 81–84.
[26] D. B. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 1992, vol. 1, pp. 25–28.
[27] F. K. Soong and E. F. Huang, "A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," in *Proc. Workshop Speech and Natural Language*, 1990, pp. 12–19.
[28] M. S. Waterman and M. Eggert, "A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons," *J. Mol. Biol.*, vol. 197, pp. 723–725, 1987.
[29] M. A. S. Saqi and M. J. E. Sternberg, "A simple method to generate non-trivial alternate alignments of protein sequences," *J. Mol. Biol.*, vol. 219, pp. 727–732, 1991.
[30] X. Huang and W. A. Miller, "A time-efficient, linear-space local similarity algorithm," *Adv. Appl. Math*, vol. 12, pp. 337–357, 1991.
[31] L. A. Mirny and E. I. Shakhnovich, "Protein structure prediction by threading. why it works and why it does not," *J Mol. Biol.*, vol. 283, pp. 507–526, 1998.
[32] J. R. Bienkowska, L. Yu, S. Zarakhovich, R. G. J. Rogers, and T. F. Smith, "Protein fold recognition by total alignment probability," *Proteins*, vol. 40, no. 3, pp. 451–462, 2000.
[33] A. Krogh, "Two methods for improving performance of an HMM and their application for gene finding," *J. Mol. Biol.*, vol. 219, pp. 727–732, 1991.
[34] S. L. Cawley and L. Pachter, "HMM sampling and applications to gene finding and alternative splicing," *Bioinform.*, vol. 19, pp. ii36–ii41, 2003.
[35] P. Fariselli, P. L. Martelli, and R. Casadio, "A new decoding algorithm for hidden markov models improves the prediction of topology of all-beta membrane proteins," *BMC Bioinform.*, vol. 6, pp. S12–S12, 2005.
[36] P. G. Bagos, T. D. Liakopoulos, I. C. Spyropoulos, and S. J. Hamodrakas, "A hidden markov model method capable of predicting and discriminating β -barrel outer membrane proteins," *BMC Bioinform.*, vol. 5, no. 29, 2004.
[37] 3rd Generation Prediction of Secondary Structure [Online]. Available: http://www.embl-heidelberg.de/~rost/Papers/1999_humana/paper.html.
[38] Z. Aydin, Y. Altunbasak, and M. Borodovsky, "Protein secondary structure prediction with semi Markov HMMS," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc.*, 2004, vol. 5, pp. 577–580.
[39] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[40] D. Nilsson and J. Goldberger, "Sequentially finding the N-best list in hidden Markov models," presented at the 17th Int. Joint Conf. Artificial Intelligence, 2001.
[41] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675–685, 1969.
[42] N. J. Nilsson, *Problem Solving Methods of Artificial Intelligence*. New York: McGraw-Hill, 1971.
[43] L. R. Bahl and F. Jelinek, "Apparatus and method for determining a likely word sequence from labels generated by an acoustic processor," U.S. Patent 4 748 670, May 1988.
[44] P. S. Gopalakrishnan, L. R. Bahl, and R. L. Mercer, "A tree-search strategy for large vocabulary continuous speech recognition," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, 1995, vol. 1, pp. 572–575.
[45] D. B. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, 1992, vol. 1, pp. 25–28.
[46] Tailbiting Decoder and Method, European Software Patents [Online]. Available: http://swpat.ffii.org/pikta/txt/ep/1258/086/#data.
[47] B. Rost, "Twilight zone of protein sequence alignments," *Protein Eng.*, vol. 12, pp. 85–94, 1999.
[48] J. A. Cuff and G. J. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," *Proteins*, vol. 34, pp. 508–519, 1999.
[49] B. Rost, C. Sander, and R. Schneider, "Redefining the goals of protein secondary structure prediction," *J. Mol. Biol.*, vol. 235, pp. 13–26, 1994.
[50] A. Zemla, C. Venclovas, K. Fidelis, and B. Rost, "A modified definition of SOV, a segment-based measure for protein secondary structure prediction assessment," *Proteins*, vol. 34, pp. 220–223, 1999.
[51] I. Ruczinski, C. Kooperberg, R. Bonneau, and D. Baker, "Distributions of β -sheets in proteins with application to structure prediction," *Proteins: Structure, Function, Genom.*, vol. 48, pp. 85–97, 2002.
[52] D. Voet and J. G. Voet, *Biochemistry, 3rd Edition*. New York: Wiley, 2004.

Zafer Aydin (S'04) received the B.S. and M.S. degrees in electrical engineering from Bilkent University, Ankara, Turkey, in 1999 and 2001, respectively. He is currently pursuing the Ph.D. degree at the Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta.

His research interests include bioinformatics, computational biology, pattern recognition, and machine learning.

Yucel Altunbasak (S'94–M'97–SM'01) received the B.S. degree (Hons.) from Bilkent University, Ankara, Turkey, in 1992 and the M.S. and Ph.D. degrees from the University of Rochester, Rochester, NY, in 1993 and 1996, respectively.

He joined Hewlett-Packard Research Laboratories (HPL), Palo Alto, CA, in 1996. His position at HPL provided him with the opportunity to work on a diverse set of research topics, such as video processing, coding and communications, multimedia streaming, and networking. He also taught digital video and signal processing courses at Stanford University, Stanford, CA, and San Jose State University, San Jose, CA, as a Consulting Assistant Professor. He joined the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, in 1999, where he is currently an Associate Professor. He is currently working on industrial- and government-sponsored projects related to multimedia networking, wireless video, video coding, genomics signal processing, and inverse imaging problems, such as super-resolution and demosaicking. His research efforts to date have resulted in more than 110 peer-reviewed publications and 15 patents/patent applications. Some of his inventions have been licensed by Office of Technology Licensing at Georgia Tech.

Dr. Altunbasak is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, *Signal Processing: Image Communications*, and for the *Journal of Circuits, Systems and Signal Processing*. He served as the lead Guest Editor on the *Image Communications* Special Issue on Wireless Video. He is the Vice President of the IEEE Communications Society Multimedia Communications Technical Committee and has been elected to the IEEE Signal Processing Society IMDSP Technical Committee. He has served as a Co-Chair for Advanced Signal Processing for Communications Symposia at ICC03, a Track Chair at ICME03 and ICME04, a Panel Sessions Chair at ITRE03, a Session Chair at various international conferences, and a Panel Reviewer for government funding agencies. He served as the Technical Program Chair for ICIP-2006. He is a coauthor for a conference paper that received the Best Student Paper Award at ICIP03 and coauthored a conference paper that has been selected as design finalist at EMBS2004. He received the National Science Foundation (NSF) CAREER Award in 2002 and is a recipient of the 2003 Outstanding Junior Faculty Award from the School of Electrical and Computer Engineering, Georgia Tech.

Hakan Erdogan (M'92) received the B.S. degree in electrical engineering and mathematics from Middle East Technical University (METU), Ankara, Turkey, in 1993 and the M.S. and Ph.D. degrees in electrical engineering systems from the University of Michigan, Ann Arbor, in 1995 and 1999, respectively.

Currently, he is an Assistant Professor at Sabanci University, Istanbul, Turkey, where he has been since 2002. He was with the Human Language Technologies group at IBM T.J. Watson Research Center, New York, from 1999 to 2002. His research interests are in developing and applying probabilistic methods and algorithms for multimedia information extraction and bioinformatics.