Prediction of Protein Beta-Sheets: Dynamic Programming versus Grammatical Approach

Yuki Kato[1], Tatsuya Akutsu[1], and Hiroyuki Seki[2]

[1] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan
{ykato,takutsu}@kuicr.kyoto-u.ac.jp
[2] Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192, Japan
seki@is.naist.jp

Abstract. Protein secondary structure prediction is one major task in bioinformatics and various methods in pattern recognition and machine learning have been applied. In particular, it is a challenge to predict β-sheet structures since they range over several discontinuous regions in an amino acid sequence. In this paper, we propose a dynamic programming algorithm for some kind of antiparallel β-sheet, where the proposed approach can be extended for more general classes of β-sheets. Experimental results for real data show that our prediction algorithm has good performance in accuracy. We also show a relation between the proposed algorithm and a grammar-based method. Furthermore, we prove that prediction of planar β-sheet structures is NP-hard.

Keywords: β-sheet, dynamic programming, formal grammar, computational complexity.

1 Introduction

Protein structure prediction is one of the central problems in bioinformatics and computational biology, and various approaches have so far been proposed. Secondary structure prediction is one of the major approaches. It asks which type of secondary structure (α-helix, β-strand, or others) each residue belongs to. Since it is a kind of classification problem, various machine learning and pattern recognition techniques have been applied, including hidden Markov models [3,16], logic programming [20], neural networks [22], stochastic tree grammars [1] and support vector machines [12]. Although the overall prediction accuracy of existing methods is around 75% [18], it is recognized that β-strand regions are more difficult to predict than α-helix regions. This discrepancy may come from the fact that β-sheet structures typically range over several discontinuous regions, whereas α-helices are continuous and thus depend more on local sequence patterns.

Protein threading is another major approach for protein structure prediction. In this approach, alignment between an input amino acid sequence and a template protein structure is computed. It is known that protein threading is

M. Chetty, A. Ngom, and S. Ahmad (Eds.): PRIB 2008, LNBI 5265, pp. 66–77, 2008.

NP-hard if pairwise interactions of residues must be taken into account [2,17]. However, several optimal algorithms have been developed for protein threading with pairwise residue-residue interactions under an assumption that insertions or deletions do not occur in core regions (i.e., α-helices and β-strands) [26]. Although it is usually overlooked in literature, there is a similarity between protein secondary structure prediction and protein threading. In protein threading (with pairwise interactions), configuration of core regions is given in advance (from a template 3D structure) and each core (α-helix or β-strand) region is searched for in an input protein sequence. In secondary structure prediction, configuration of core regions is not given in advance and each residue is assigned to one of the three classes of secondary structures.

Although we have discussed about protein structure prediction, RNA secondary structure prediction is another important problem in bioinformatics and computational biology. One of the common approaches of RNA secondary structure prediction is use of (stochastic) grammars, which include stochastic context-free grammar [10,23], stochastic multiple context-free grammar [15], parallel communicating grammar [7], crossed-interaction grammar [21] and tree adjoining grammar [25]. These grammars may also be useful to model other pattern recognition problems.

Recently, Chiang et al. [8] proposed some grammar-based methods for protein secondary structure prediction. In particular, they proposed use of *range concatenation grammar* (RCG) [5] for β-sheet modeling. They suggested that linearly ordered β-sheets can be modeled by using a simple RCG and can be predicted in $O(n^5)$ time, where n is the number of residues in a given protein sequence. They also suggested that β-barrels and more complex β-sheet structures can be modeled by using RCG, while the time complexity increases to $O(n^7) \sim O(n^{12})$ depending on the complexity of β-sheet structures. However, they did not show how to incorporate residue-residue interaction preferences into the RCG-based methods. Furthermore, they posed the following question for proving NP-hardness of β-sheet prediction: "it remains to be seen whether such dependencies might be needed, for example, in calculating conformation counts for β-sheets."

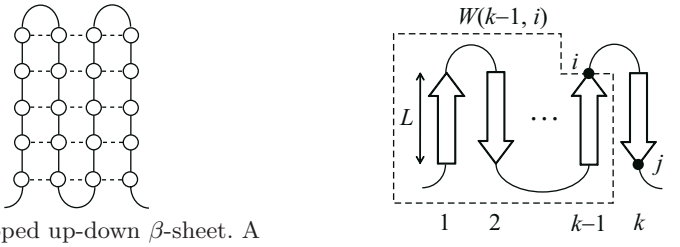
In this paper, we propose a simple and flexible dynamic programming algorithm for prediction of antiparallel up-down β-sheets. This algorithm is based on RCG approach [8], where no experimental results on structure prediction were provided. It is noteworthy that our method explicitly takes pairwise interaction preferences into account and thus can be applied to real protein sequences. Hubbard [13] also used interstrand residue pairing preferences to predict β-strand contact maps, but did not show an original prediction algorithm specific for β-sheet prediction. Our prediction algorithm achieved good performance of overall per-residue accuracy $Q_3 \approx 80\%$ for nonhomologous protein sequences with up-down topology, where there are only two secondary structural states. Although types of β-sheet structures that can be handled by our method are restricted, the technique is extensible to more complex β-sheet structures including β-barrel. We also provide insight into an existing grammar-based method. Furthermore,

we show that prediction of planar β-sheet structures is NP-hard. This result gives an answer to the question posed by Chiang et al. [8].

2 Methods

2.1 Ungapped Antiparallel β-Sheet

β-sheets are formed by pairwise interaction of several (consecutive) amino acids, called β-strands, in parallel and/or antiparallel way. Antiparallel β-structure is a fundamental topology of β-sheet, and many proteins include it in their domain. Although there are a large number of combinations of β-strands, it is known that the number of topologies of the class of antiparallel β-sheets is relatively few [6]. In this section, we are concerned with the simplest topology among them, called *up-down β-sheet*, where all strands have antiparallel topology via hydrogen bonding and they are connected by hairpin. In addition, suppose that every amino acid of β-strands is involved in hydrogen bonding, which we call *ungapped β-sheet*. Fig. 1 (a) illustrates an ungapped up-down β-sheet. This assumption enables us to design more efficient prediction algorithm in terms of computational complexity.



(a) An ungapped up-down β-sheet. A white circle represents an amino acid and a dashed line indicates a hydrogen bond.

(b) A schematic diagram of the dynamic programming algorithm

Fig. 1. Illustration of an ungapped up-down β-sheet

Let $a = a_1 a_2 \cdots a_n$ denote an amino acid sequence to be analyzed. We consider an ungapped up-down β-sheet that have N strands of the same length L where $N \leq \lfloor \frac{n}{L} \rfloor$. The reason why we can assume L is fixed is that we are concerned with only ungapped β-sheets. Because of this assumption, a β-sheet can be represented by an N-tuple of the start positions of β-strands $(p_1, p_2, \ldots, p_N)$ in the amino acid sequence a. Note that $p_i + L \leq p_{i+1}$ must be satisfied to prevent adjacent strands from overlapping each other. Let $s : (a_i, a_j) \to \mathbb{R}$ be a score (energy) function between two amino acid residues. Then, the ungapped up-down β-sheet prediction problem can be defined as follows:

Definition 1. ***(Ungapped up-down β-sheet prediction problem)***
Input: *An amino acid sequence $a = a_1 a_2 \cdots a_n$, the number of strands N, their common length L and a score function s.*

Output: *An ungapped up-down β-sheet $(p_1, p_2, \ldots, p_N)$ that minimizes the following score:*

$$\sum_{i=1}^{N-1} \sum_{j=1}^{L} s(a_{p_i+j-1}, a_{p_{i+1}+L-j}),$$

subject to $p_i + L \leq p_{i+1}$ $(i = 1, 2, \ldots, N)$.

2.2 Dynamic Programming Algorithm

We provide a dynamic programming (DP) algorithm for predicting ungapped up-down β-sheets. In the experiments described later, we predict β-sheet by changing the value of N, though N is fixed in the algorithm described below. Let $W(k, j)$ be the minimum free energy of up-down β-sheet for $a_1 \cdots a_j$, where j is the last position of the kth β-strand (see Fig. 1 (b)). $W(k, j)$ can be calculated by the following simple recursion formula:

$$W(k, j) = \min_i \{W(k-1, i) + S(i, j, L)\},$$

where

$$S(i, j, L) = \sum_{h=1}^{L} s(a_{i-L+h}, a_{j-h+1}).$$

The detailed description of the DP algorithm is presented below.

Initialization:
for $j = L$ **to** n **do** $W(1, j) = 0$.
Recursion:
for $k = 2$ **to** N **do**
for $j = kL$ **to** n **do**

$$W(k, j) = \min_{(k-1)L \leq i \leq j-L-2} \{W(k-1, i) + S(i, j, L)\}.$$

Note that this algorithm takes the length of hairpin into consideration by restricting the range of i in the recursive step.

A simple inspection of the recursive step yields the time complexity of the algorithm. Since the double "for loop" takes $O(n^2)$ time and the minimum operation takes $O(n)$ time, the time complexity is evaluated as $O(n^3)$. Obviously, the algorithm requires $O(n^2)$ space. Note that the optimal β-sheet itself can be constructed by a simple traceback procedure.

Although our DP algorithm can only handle up-down β-sheets, we can easily extend our method to predict more complicated structures, including consecutive parallel β-sheets, β-barrels as well as gapped structures.

In order to extend the algorithm for β-barrels, we compute the following:

$$W(k, j, i_0) = \min_i \{W(k-1, i, i_0) + S(i, j, L)\}$$

for each i_0 under the condition that

$$W(1, j, i_0) = \begin{cases} 0 \ \ (j = i_0), \\ \infty \ \text{(otherwise)}. \end{cases}$$

Then, we compute the minimum of

$$W(N, j, i_0) + \sum_{h=1}^{L} s(a_{i_0-L+h}, a_{j-h+1}).$$

In this case, the time complexity increases from $O(n^3)$ to $O(n^4)$. More complex β-sheet structures may be treated by using the divide-and-conquer approach proposed by Xu et al. [26]. However, the time complexity would increase as the complexity of β-sheet increases as suggested by the NP-hardness result in Section 5.

In order to extend the algorithm for gapped antiparallel β-sheets, it is enough to modify the definition of $S(i, j, L)$ so that it denotes the score of an optimal *alignment* between $a_{i-L+1} \cdots a_i$ and $a_j \cdots a_{j-L+1}$. In this case, the total time complexity increases to $O(n^4)$. Of course, we can extend it for prediction of gapped β-barrels. In that case, the time complexity remains $O(n^4)$. Capability of handling gapped β-sheets is one of the big advantages of our proposed method since gaps in core regions are not allowed in protein threading with residue-residue pairwise interactions [26].

3 Experimental Results

3.1 Data

In our experiments on prediction of up-down β-sheets with β-barrels, we used real protein sequences with known structure available in PDB_SELECT (2007) [11] as the test sets (see Table 1). The criteria for selecting test data are as follows:

(1) The test sequences are contained in the 25% threshold list of PDB_SELECT, where no two proteins have more than 25% sequence identity.
(2) They have at least four β-strands specified in DSSP [14]. Note that we do not count a residue involved in an isolated β-bridge as one strand.
(3) All but at most one pair of adjacent β-strands in the primary sequence are involved in hydrogen bonding. This constraint results from lack of a perfect set of up-down β-sheets in the list.

3.2 Tests

Since the sequences selected above actually have different strand lengths, we set the strand length constant L by rounding the mean of their actual lengths. We used a contact potential table derived from 785 proteins described in [9] as

Table 1. Accuracy of antiparallel β-sheet prediction

(a) Up-down β-sheet prediction

PDBID	N	n	L	Q_3 [%]	Q_E [%]	Q_E^{pred} [%]
2B9K	4	47	7	72.34	77.78	75.00
1AUU	4	55	4	83.64	70.59	75.00
1NY4	4	82	6	84.15	72.00	75.00
1TPN	5	50	4	68.00	61.11	55.00
2E6Z	5	59	4	74.58	61.90	65.00
2DIG	5	68	5	82.35	74.07	80.00
2JN4	6	66	5	87.88	89.29	83.33
2BT9	8	90	8	80.00	88.33	82.81
Average				79.12	74.38	73.89

(b) β-barrel prediction

PDBID	N	n	L	Q_3 [%]	Q_E [%]	Q_E^{pred} [%]
1Q9F	7	148	10	70.95	69.01	70.00
1MM4	8	170	9	64.71	58.11	59.72
1G90	8	176	11	82.39	81.32	84.09
1FW3	12	269	12	63.20	65.73	65.28
1PHO	16	330	11	66.36	67.96	69.89
Average				69.52	68.43	69.80

the score function s. Implementation of the prediction algorithms for up-down β-sheet and β-barrel was carried out in Java (version 1.6.0_03) on a machine with Intel Core2 CPU 6700 2.66GHz, 1.57GHz and 2.99GB RAM. To evaluate prediction accuracy of our algorithms, we measured per-residue accuracy Q_3, Q_E and Q_E^{pred}. Q_3 is the ratio of correctly predicted residues in overall secondary structural elements. Note that there are only two secondary structural states in this case (i.e., strand and other), and observed structures that we referred to are specified in DSSP. Q_E is defined as the ratio of the number of correctly predicted residues of the β-strands to the total number of residues of the strands in the observed structure, which corresponds to sensitivity. Q_E^{pred}, corresponding to specificity, is the ratio of the number of correctly predicted residues of the β-strands to the total number of predicted residues of the strands. Prediction results on up-down β-sheet prediction are shown in Table 1 (a) and Fig. 2, and results on β-barrel prediction are shown in Table 1 (b). Computation time of up-down β-sheet prediction was 0.19 seconds on average, whereas computation time of β-barrel prediction was 480.04 seconds on average. Note that this discrepancy arises from the difference of time complexity (i.e., $O(n^3)$ vs. $O(n^4)$).

Observed beta sheet (E: extended strand, participates in beta ladder):
MKVMIRKTATGHSAYVAKKDLEELIVEMENPALWGGKVTLANGWQLELPAMAADTPLPITVEARKL
..EEEEE....EEEEE....EEEEEEEE........EEEE....EEE...........EEE.....

Predicted beta sheet:
MKVMIRKTATGHSAYVAKKDLEELIVEMENPALWGGKVTLANGWQLELPAMAADTPLPITVEARKL
..EEEEE....EEEEE.......EEEEE.......EEEEE...EEEEE........EEEEE.....

Fig. 2. Comparison of the observed structure with the predicted one for 2JN4. Underlined residues indicate that they agree with correct residues of the β-strands.

3.3 Discussion

Experimental results on up-down β-sheet prediction show that our prediction algorithm has good performance in accuracy for several real protein sequences. One reason for high accuracy is that the contact potentials computed in [9] are good in quality. In fact, we performed the same prediction tests using other contact potentials presented in [4,24,27], where average prediction accuracy is 76.62% in Q_3, 71.71% in Q_E and 71.52% in Q_E^{pred} for [4], 72.52% in Q_3, 66.14%

in Q_E and 65.68% in Q_E^{pred} for [24], and 67.86% in Q_3, 60.01% in Q_E and 59.73% in Q_E^{pred} for [27]. These values are lower than the average accuracy when using the contact potentials in [9]. It should be noted that a few protein structures (1AUU and 1TPN) used to compute the contact potentials in [9] were also used for our experiments. However, most accuracy assessment for these two proteins is lower than the average (see Table 1 (a)), and there seems to be no positive bias that improves the accuracy of the algorithm.

It can be seen that the choice of the number of β-strands N is important to achieve good prediction accuracy. After we performed the test shown in Table 1 where N was actually chosen as the observed number of strands N_{obs}, we developed a simple method of selecting N during computation of the DP table W. More specifically, we calculated the average of $W(k, j)$ for each k ($2 \leq k \leq \lfloor \frac{n}{L} \rfloor$), denoted by $W_{avg}(k)$, and then selected N as the first k such that $W_{avg}(k) < W_{avg}(k+1)$ holds while calculating in an increasing order of k. Although the average prediction accuracy for up-down β-sheets drops to 72.52% in Q_3, 74.41% in Q_E and 65.17% in Q_E^{pred}, the value N determined by this method ranges from $N_{obs}-1$ to $N_{obs}+2$, which shows a relatively good tendency in choice of N.

As Table 1 (b) indicates, prediction accuracy for β-barrels is not so good as compared with the results on up-down β-sheet prediction. This may suggest that achieving good accuracy is difficult if the topology of the β-sheet to be analyzed becomes complex. To achieve higher accuracy than the present accuracy for β-barrels, it would be interesting to incorporate "torsion changes" into our algorithms, which is considered to be important for the stability of a protein.

As compared to another approach for β-sheet prediction, accuracy of a method using *ranked node rewriting grammar* (RNRG) [1] is roughly 74% in Q_E, which is comparable to the performance of our method. Although the test data used in our experiments are different from the data used in the RNRG-based method, we tested more sequences than they did. Furthermore, it should be noted that we never used a training algorithm to estimate score parameters, whereas the RNRG approach performed training of probability parameters using an inside-outside algorithm, which is prohibitively time-consuming.

4 Remarks on Grammatical Modeling

4.1 Definitions

Range concatenation grammar [5] is defined as a deductive system on sequences. A (positive) range concatenation grammar (RCG) is a 5-tuple $G = (N, T, V, P, S)$, where N, T, V and P are finite sets of predicate names, terminals, variables, rules, respectively, and $S \in N$ is the start predicate. For each predicate name $A \in N$, a nonnegative integer $\dim(A)$ is specified. Each rule in P has the shape $\psi_0 \rightarrow \psi_1 \cdots \psi_k$. This rule means that ψ_0 holds when all of $\psi_1, \ldots, \psi_k$ hold. Each ψ_i ($0 \leq i \leq k$) in the rule is a predicate of the shape $A_i(\alpha_{i1}, \ldots, \alpha_{i\,\dim(A_i)})$, where $A_i \in N$

and each α_{ij} $(1 \leq j \leq \dim(A_i))$ is just a variable in V if $1 \leq i \leq k$. The following is a simple example of rules:

$$S(xyz) \to A(x,y)B(z), \qquad A(axb,cyd) \to A(x,y), \qquad B(ez) \to B(z),$$
$$A(ab,cd) \to \varepsilon, \qquad B(\varepsilon) \to \varepsilon.$$

Let $\Rightarrow$ denote the one-step derivation relation. For example,

$$S(aabbccdde) \Rightarrow A(aabb,ccdd)B(e) \Rightarrow A(ab,cd)B(e) \Rightarrow B(e) \Rightarrow B(\varepsilon) \Rightarrow \varepsilon.$$

Let $\stackrel{+}{\Rightarrow}$ denote the transitive closure of $\Rightarrow$. The language generated by an RCG G is defined as $L(G) = \{w \mid S(w) \stackrel{+}{\Rightarrow} \varepsilon\}$. For the above example, $L(G) = \{a^m b^m c^m d^m e^n \mid m \geq 1, n \geq 0\}$. We also say that A *generates* w when $A(w) \stackrel{+}{\Rightarrow} \varepsilon$.

If every variable occurs at most once in the left-hand side (rsp. right-hand side) of a rule, the rule is called *left linear* (rsp. *right linear*). For example, $S(x) \to S_1(x)S_2(x)$ is left linear but not right linear.

4.2 Modeling by RCG

Chiang et al. [8] presented the following RCG to generate linearly ordered β-sheets:

$$Beta(xy) \to B(x,y), \qquad B(xyz,y') \to B(x,y)Adj(y,y'),$$
$$B(yz,y') \to Adj(y,y'),$$
$$Adj(x,y) \to Anti(x,y), \qquad Adj(x,y) \to Par(x,y),$$
$$Anti(ax,y\overline{a}) \to Anti(x,y), \qquad Anti(\varepsilon,\varepsilon) \to \varepsilon,$$
$$Par(ax,\overline{a}y) \to Par(x,y), \qquad Par(\varepsilon,\varepsilon) \to \varepsilon,$$

where $a, \overline{a} \in T$ stand for amino acid residues that are connected with each other by hydrogen bond. (We extend the notion $\overline{u}$ for a sequence u.) Par and $Anti$ generate parallel and antiparallel strands, respectively. $B(u,v)$ means that uv is a β-sheet where the second argument v is the "last" strand. Thus, the second rule says that if xy is a β-sheet (with y the last strand) and (y,y') constitutes a pair of adjacent strands, then $xyzy'$ is also a β-sheet (with y' the last strand) for an unpaired subsequence z. In this rule, the right nonlinearity plays a crucial role that expresses the constraints that the last strand y should be one component y of pair strands (y,y'). The time complexity of the structure prediction based on parsing of RCG is easily derived by counting the independent positions that appear in the arguments of the left-hand side for each rule and taking the maximum of them. For example, the independent positions are marked by $*_i$ $(1 \leq i \leq 5)$ for the second rule as $B(*_1 x_{*_2} y_{*_3} z_{*_4}, y'_{*_5})$. This is the maximum among all the above rules, thus the complexity is $O(n^5)$ where n is the length of an input sequence.

Returning to the problem of this paper, we assume that the length of each strand is L. This means that $|y| = |y'| = L$ in the second rule, implying that

the position $*_3$ and $*_5$ is determined by $*_2$ and $*_4$, respectively. Thus, the time complexity becomes $O(n^3)$, which is the same order as our algorithm for up-down β-sheet in Section 2. Note that the formalism in [8] does not incorporate residue-residue interaction preferences. Implementation or experimental results on β-sheet prediction based on RCG has not been reported as far as the authors know. On the other hand, we have performed experiments with real protein sequences. Although our algorithms currently consider only antiparallel β-sheets, it is not difficult to extend our proposed algorithms so that parallel structures can be treated, as described in Section 2.2.

5 Hardness Result

Although we have presented an $O(n^3)$ time dynamic programming algorithm in Section 2, it remains a question whether *generalized* ungapped β-sheets can be predicted in polynomial time or not. To discuss the complexity of such a prediction problem, we define the corresponding decision problem as follows:

Definition 2. ***(Ungapped β-sheet prediction problem, UGBETA)***
Input: *An amino acid sequence, a topology diagram and a real number e.*
Output: *"Yes" if and only if there exists an ungapped β-sheet with some free energy e or less.*

In the following, we will show that UGBETA is NP-complete by reducing the longest common subsequence problem that is known to be NP-complete [19]:

Definition 3. ***(Longest common subsequence problem, LCS)***
Input: *m sequences over an alphabet and a positive integer k.*
Output: *"Yes" if and only if there exists a common subsequence of length k or more, which is not necessarily consecutive.*

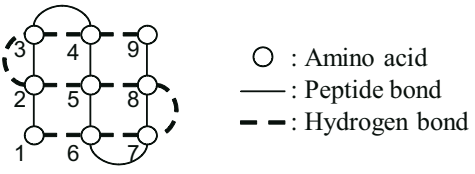
Theorem 1. *UGBETA is NP-complete even if the topology diagram is planar.*

Proof. Assume that each β-strand consists of exactly one amino acid (i.e., $L = 1$) (see Fig. 3). We can also show that NP-completeness result holds for $L \geq 2$.

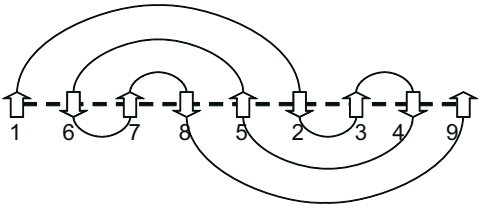
First, it is easy to see that UGBETA belongs to $\mathcal{NP}$. Guess an ungapped β-sheet from the amino acid sequence, and check that it has at most e of free energy value under some energy function.

Next, let us show how to reduce LCS to UGBETA for the proof of NP-hardness. Let $w_1, w_2 \ldots, w_m \in \{0,1\}^*$ be instance sequences of LCS. Without loss of generality, we assume that m is an even number. If it is odd, we simply add a new sequence w_{m+1} that is the same as w_m. Also, we can assume that a positive integer k is an odd number. If it is even, we simply add 0 at the end of each w_i $(i = 1, 2, \ldots, m)$. We construct from $w_1, w_2, \ldots, w_m$ an amino acid sequence $A = B_0 B_1 B_2 \cdots B_m B_{m+1} \in \{0, 1, x, y\}^*$, where

$$B_0 = x(xyx)^{(k+1)/2}y, \qquad B_{2i-1} = xw_{2i-1}xy \quad (i = 1, 2, \ldots, m/2),$$
$$B_{2i} = xw_{2i}^R xy \quad (i = 1, 2, \ldots, m/2), \qquad B_{m+1} = x(xyx)^{(k+1)/2}$$



(a) A β-sheet from the viewpoint of sequence



(b) The topology diagram of β-sheet of (a)

Fig. 3. A simplified ungapped β-sheet ($L = 1$). For simplicity of illustration, we allow hydrogen bond to be compatible with peptide bond.

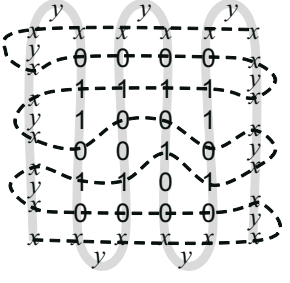


Fig. 4. Example of an amino acid sequence over $\{0, 1, x, y\}$ constructed from an LCS instance, where $w_1 = 011010$, $w_2 = 010010$, $w_3 = 010100$, $w_4 = 011010$, $m = 4$ and $k = 5$.

(see Fig. 4). Note that w_i^R denotes the reverse sequence of w_i, and $B_{i,j}$ that will be used below denotes the jth symbol of B_i. The score (energy) function s is defined in such a way that $s(0,0) = s(1,1) = -1$, $s(x,x) = -\alpha$ where α is set at some positive constant times nm, and defined as 0 for the other pairs. It is obvious that this transformation can be accomplished in polynomial time. Then, we must show the following:

– There exists a common subsequence of length k in $w_1, w_2, \ldots, w_m$ if and only if there exists an ungapped β-sheet of A with free energy $-k(m + \alpha - 1) - \alpha(2m + 3)$.

We omit a detailed proof of the above statement in this version as space is limited. It should be noted that the topology diagram used in this proof is planar (see Fig. 3 (b)). □

6 Concluding Remarks

We presented dynamic programming algorithms for predicting ungapped up-down β-sheet and its extensions. Experimental results on ungapped up-down β-sheet prediction showed that performance is good enough to distinguish β-sheet regions from non-β-sheet ones. However, we have not presented complete comparison with other models for β-sheet prediction, which is left as our future work.

Computational models that predict biomolecule structure with high accuracy are needed in bioinformatics. When we develop a model for prediction, it is important to assign some biologically appropriate score to the model. In our experiments using the dynamic programming algorithms, we used contact potentials and did not perform training from the sequence sets. It might be possible to design a training algorithm based on the EM algorithm, in which case, the prediction accuracy would be higher. If we choose a grammatical approach, training has to be carried out due to the difficulty in assigning optimal probabilities.

As shown in Section 5, arbitrary ungapped planar β-sheet prediction is NP-hard. However, this claim does not always imply that efficient algorithms never exist for small input sets. Most protein sequences consist of at most a few hundred amino acid residues, and there is room for further investigation into the development of efficient algorithms even if topologies that we wish to handle are complex. Furthermore, it is a challenging task to develop an efficient algorithm for predicting protein structures that include the combination of α-helix and β-sheet.

References

1. Abe, N., Mamitsuka, H.: Predicting Protein Secondary Structure Using Stochastic Tree Grammars. Machine Learning 29, 275–301 (1997)
2. Akutsu, T., Miyano, S.: On the Approximation of Protein Threading. Theor. Comp. Sci. 210, 261–275 (1999)
3. Asai, K., Hayamizu, S., Handa, K.: Prediction of Protein Secondary Structure by the Hidden Markov Model. Bioinformatics 9, 141–146 (1993)
4. Berrera, M., Molinari, H., Fogolari, F.: Amino Acid Empirical Contact Energy Definitions for Fold Recognition in the Space of Contact Maps. BMC Bioinformatics 4 (2003)
5. Boullier, P.: Range Concatenation Grammars. In: Sixth Intl.Workshop on Parsing Technologies (IWPT 2000), pp.53–64 (2000)
6. Branden, C., Tooze, J.: Introduction to Protein Structure, 2nd edn. Garland Publishing (1999)
7. Cai, L., Malmberg, R.L., Wu, Y.: Stochastic Modeling of RNA Pseudoknotted Structures: A Grammatical Approach. Bioinformatics 19, i66–i73 (2003)
8. Chiang, D., Joshi, A.K., Searls, D.B.: Grammatical Representations of Macromolecular Structure. J. Comp. Biol. 13, 1077–1100 (2006)
9. Dosztányi, Z., Csizmók, V., Tompa, P., Simon, I.: The Pairwise Energy Content Estimated from Amino Acid Composition Discriminates between Folded and Intrinsically Unstructured Proteins. J. Mol. Biol. 347, 827–839 (2005)
10. Eddy, S.R., Durbin, R.: RNA Sequence Analysis Using Covariance Models. Nucl. Acids Res. 22, 2079–2088 (1994)

11. Hobohm, U., Scharf, M., Schneider, R., Sander, C.: Selection of a Representative Set of Structures from the Brookhaven Protein Data Bank. Protein Sci. 1, 409–417 (1992)
12. Hua, S., Sun, Z.: A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach. J. Mol. Biol. 308, 397–407 (2001)
13. Hubbard, T.J.P.: Use of β-Strand Interaction Pseudo-Potentials in Protein Structure Prediction and Modelling. In: The Twenty-Seventh Annual Hawaii Intl. Conf. on System Sciences, pp. 336–344 (1994)
14. Kabsch, W., Sander, C.: Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Biopolymers 22, 2577–2637 (1983)
15. Kato, Y., Seki, H., Kasami, T.: RNA Pseudoknotted Structure Prediction Using Stochastic Multiple Context-Free Grammar. IPSJ Trans. Bioinformatics 47, 12–21 (2006)
16. Krogh, A., Brown, M., Mian, I.S., Sjölander, K., Haussler, D.: Hidden Markov Models in Computational Biology: Applications to Protein Model. J. Mol. Biol. 235, 1501–1531 (1994)
17. Lathrop, R.H.: The Protein Threading Problem with Sequence Amino Acid Interaction Preferences is NP-Complete. Protein Eng. 7, 1059–1068 (1994)
18. Lin, K., Simossis, V.A., Taylor, W.R., Heringa, J.: A Simple and Fast Secondary Structure Prediction Method Using Hidden Neural Networks. Bioinformatics 21, 152–159 (2005)
19. Maier, R.: The Complexity of Some Problems on Subsequences and Supersequences. J. ACM 25, 322–336 (1978)
20. Muggleton, S., King, R., Sternberg, M.: Protein Secondary Structure Prediction Using Logic-Based Machine Learning. Protein Eng. 5, 647–657 (1992)
21. Rivas, E., Eddy, S.R.: The Language of RNA: A Formal Grammar that Includes Pseudoknots. Bioinformatics 16, 334–340 (2000)
22. Rost, B., Sander, C.: Prediction of Protein Secondary Structure at Better than 70% Accuracy. J. Mol. Biol. 232, 584–599 (1993)
23. Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjölander, K., Underwood, R.C., Haussler, D.: Stochastic Context-Free Grammars for tRNA Modeling. Nucl. Acids Res. 22, 5112–5120 (1994)
24. Tanaka, S., Scheraga, H.A.: Medium- and Long-Range Interaction Parameters between Amino Acids for Predicting Three-Dimensional Structures of Proteins. Macromolecules 9, 945–950 (1976)
25. Uemura, Y., Hasegawa, A., Kobayashi, S., Yokomori, T.: Tree Adjoining Grammars for RNA Structure Prediction. Theor. Comp. Sci. 210, 277–303 (1999)
26. Xu, Y., Xu, D., Uberbacher, E.C.: An Efficient Computational Method for Globally Optimal Threading. J. Comp. Biol. 5, 597–614 (1998)
27. Zhang, C., Kim, S.H.: Environment-Dependent Residue Contact Energies for Proteins. PNAS 97, 2550–2555 (2000)