PAPER Special Section on Discrete Mathematics and Its Applications

Dynamic Programming and Clique Based Approaches for Protein Threading with Profiles and Constraints*

Tatsuya AKUTSU[†a)], **Morihiro HAYASHIDA**[†], ***Members***, **Dukka BAHADUR K.C.**[†], ***Nonmember***, **Etsuji TOMITA**[††], ***Fellow***, **Jun'ichi SUZUKI**[††], ***and*** **Katsuhisa HORIMOTO**[†††], ***Nonmembers***

SUMMARY The protein threading problem with profiles is known to be efficiently solvable using dynamic programming. In this paper, we consider a variant of the protein threading problem with profiles in which constraints on distances between residues are given. We prove that protein threading with profiles and constraints is NP-hard. Moreover, we show a strong hardness result on the approximation of an optimal threading satisfying all the constraints. On the other hand, we develop two practical algorithms: CLIQUETHREAD and BBDPTHREAD. CLIQUETHREAD reduces the threading problem to the maximum edge-weight clique problem, whereas BBDPTHREAD combines dynamic programming and branch-and-bound techniques. We perform computational experiments using protein structure data in PDB (Protein Data Bank) using simulated distance constraints. The results show that constraints are useful to improve the alignment accuracy of the target sequence and the template structure. Moreover, these results also show that BBDPTHREAD is in general faster than CLIQUETHREAD for larger size proteins whereas CLIQUETHREAD is useful if there does not exist a feasible threading.

key words: *maximum edge weight clique, dynamic programming, protein threading, profiles, distance constraints*

1. Introduction

Prediction of protein structures using computational tools is one of the important problems in computational biology. Recently, some advances have been achieved by the significant utilization of distance restraints. Xu et al. showed that (partial) information obtained from NMR experiments is useful to improve the accuracy of the protein threading method [1], where *protein threading* is one of the powerful computational approaches to protein structure prediction. Young et al. recently developed a novel experimental method to aid in construction of a homology model by using chemical cross-linking and time-of-flight (TOF) mass spectrometry to identify LYS-LYS cross-links [5]. Instead of X-ray crystallography and NMR that require much amount of pure analyte and much time for experiments, the distance restraints obtained by the intramolecular cross-links and mass spectrometry were useful to improve the accuracy. Therefore, development of algorithms in which distance constraints can be taken into account is important.

Xu et al. modified the PROSPECT algorithm for protein threading with *pairwise contact energy* and constraints [1]. Another algorithm was also proposed for the improvement in the fold recognition of the protein threading [2]. An algorithm using unassigned NMR data that relies on ROSETTA and a Monte Carlo procedure for the generation of low resolution protein structures has been developed [3]. Moreover, methods like TOUCHSTONEX [4] have been proposed that incorporates a limited number of distance restraints into the force field as NOE-specific pairwise interaction to predict protein structures at low-medium resolution. However, all of these algorithms use complicating sampling schemes and/or scoring functions.

On the other hand, *threading with profiles* (or threading with *position specific score matrices*) is also known as a powerful method for structure prediction. In particular, PSI-BLAST is widely used both for homology search and structure prediction [6]. Thus, it is reasonable to try to develop algorithms for protein threading with profiles and constraints, which may be useful to improve the prediction accuracy by PSI-BLAST. In this paper, we therefore study threading with profiles and constrains in terms of both theoretical and practical aspects.

At first, we show that finding a *feasible threading* (i.e., a threading that satisfies all the constraints derived from experiments) is NP-hard. Moreover, we show a strong hardness result on the approximation of an optimal feasible threading. It should be noted that protein threading with pair score functions is NP-hard [7], [8], whereas protein threading with profiles can be solved efficiently using dynamic programming as in sequence alignment. Our results show that adding constraints makes the problem much harder.

Furthermore, we develop two practical exact algorithms for protein threading with profiles and constraints: CLIQUETHREAD and BBDPTHREAD. CLIQUETHREAD reduces constrained threading to the maximum edge weight clique problem. Though the maximum clique is NP-hard, several practically efficient algorithms have been developed [9], [10]. BBDPTHREAD combines a DP (dynamic programming) algorithm and a branch-and-bound procedure, where the DP algorithm is developed

Manuscript received August 13, 2005.
Manuscript revised November 2, 2005.
Final manuscript received December 15, 2005.

[†]The authors are with the Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji-shi, 611-0011 Japan.
[††]The authors are with the Graduate School of Electro-communications, The University of Electro-Communications, Chofu-shi, 182-8585 Japan.
[†††]The author is with the Human Genome Center, Institute of Medical Science, The University of Tokyo, Tokyo, 108-8639 Japan.

*A preliminary version of the paper was presented at IEEE 4th Symp. Bioinformatics and Bioengineering (BIBE2004).

a) E-mail: takutsu@kuicr.kyoto-u.ac.jp

DOI: 10.1093/ietfec/e89–a.5.1215

based on our previous work [7]. We perform computational experiments on both algorithms using PDB (Protein Data Bank) data [11]. The results suggest that constraints are useful to improve the prediction accuracy of protein threading. That is, constraints are useful to improve the quality of alignments between sequences and structures.

From a theoretical viewpoint, there exist several related works. The authors previously studied approximation algorithms for protein threading with pairwise contact energy [7]. As discussed in Sect. 6, threading with profiles and constraints is very similar to that problem. Goldman et al. studied theoretical aspect of protein structure alignment [12]. Recently, the longest common subsequence problem with arc annotations is extensively studied [13]–[16]. Though these studies have been done independently, similar results were obtained. However, it should be pointed out that our previous work [7] is one of the earliest work.

Although, the authors have already presented the improved version of the clique based algorithm for protein threading with profiles and constraints [17], [18], only this paper deals with the theoretical analysis of the problem, and the dynamic programming based algorithm, BBDPTHREAD. Furthermore, the clique based algorithm is also initially presented in the preliminary version of this paper. It is also to be noted that the preliminary version of this paper appeared much before the improved version of the clique based algorithm [17], [18]. Hence, this paper originally presents the theoretical analysis of the protein threading with the clique based algorithm CLIQUETHREAD and the DP based algorithm BBDPTHREAD.

From a practical viewpoint, it seems that most exact algorithms for protein threading with pairwise contact energy [1], [19], [20] can be modified for threading with constraints as in [1]. However, gaps in core regions are not allowed in these algorithms whereas gaps are allowed in our algorithms. At least, our algorithms are simpler than existing algorithms and easy to implement and modify.

The organization of the paper is as follows. We begin with the formal definitions of the problems. Next, we give hardness results. Then, we present two practical algorithms and describe the results of computational experiments on these algorithms. Finally, we conclude with future work.

2. Definitions

In this section, we formally define the problems. It should be noted that the same definitions are also given in our related paper [18]. First we define a threading, where a threading (without constraint) is almost the same as a pairwise alignment here. Let $s = s_1 \dots s_m$ be a protein sequence, over an alphabet Σ, where Σ is the set of amino acids (i.e., $|\Sigma| = 20$). We also use s_i to denote the position of s_i in s. Let $t = t_1 \dots t_n$ be a template protein structure, where t_i is a residue (or the position of t_i) in t. t can be considered as a sequence of C^α (or C^β) atoms of the protein structure. A *threading* between s and t is obtained by inserting *gap symbols* ('$-$') into or at either end of s and t such that the resulting sequences s' and t' are of the same length l, where it is not allowed for each $i \leq l$ that both s'_i and t'_i are gap symbols.

For each template structure t, a *profile* PF_t is assigned, where PF_t is a function from $(\Sigma \cup \{-\}) \times \{t_1, \dots, t_n, -\}$ to the set of real numbers $\mathcal{R}$. Though affine gap costs are not represented by this notation, affine gap costs are also taken into account in the algorithms in this paper. The *score of a threading* (s', t') is defined by $\sum_{i=1}^{l} PF_t(s'_i, t'_i)$.

Problem 1 (Profile Threading without Constraint): Given s, t and PF_t, find a threading (s', t') with the maximum score.

Next, we define constraints. We write $\psi(s_i) = t_j$ if s_i and t_j are aligned into the same column in a threading (s', t'). If s_i is aligned with the gap symbol, we define $\psi(s_i) =$'$-$.' For a target sequence s, an *arc set* A_s is associated, which is a set of pairs of positions of s and each pair $(s_i, s_{i'}) \in A_s$ must satisfy $1 \leq i < i' \leq m$. Similarly, A_t denotes an arc set for a template structure t. In this paper, s_i appearing in A_s must not be aligned with a gap symbol at the same column.

For each pairs $(s_i, s_{i'})$ and $(t_j, t_{j'})$, $IC(s_i, s_{i'}, t_j, t_{j'}) = 0$ if these pairs satisfy a constraint on $(s_i, s_{i'})$. If $(s_i, s_{i'}) \notin A_s$, $IC(s_i, s_{i'}, t_j, t_{j'}) = 0$, otherwise (i.e., the pairs do not satisfy a constraint, or $(s_i, s_{i'}) \in A_s$ but $(t_j, t_{j'}) \notin A_t$), $IC(s_i, s_{i'}, t_j, t_{j'}) = 1$. It should be noted that IC means *inconsistency*. Though IC is defined in a general way as above, we employ the following definition in the practical algorithms: $IC(s_i, s_{i'}, t_j, t_{j'}) = 0$ if $|dist(s_i, s_{i'}) - dist(t_j, t_{j'})|$ is less than a threshold Θ, where $dist(s_i, s_{i'})$ (resp. $dist(t_j, t_{j'})$) denotes the distance between positions of C^α (or C^β) atoms associated with s_i and $s_{i'}$ (resp. t_j and $t_{j'}$), and A_s and A_t correspond to the sets of known distances. As showin in Sect. 5, this type of constraint is useful to improve the prediction accuracy of protein threading.

We consider two types of constrained threading problems.

Problem 2 (Profile Threading with Strict Constraints. See Fig. 1): Given (s, A_s), (t, A_t), PF_t, and IC, find a threading (s', t') with the maximum score under the condition that $IC(s_i, s_{i'}, \psi(s_i), \psi(s_{i'})) = 0$ for all $(s_i, s_{i'}) \in A_s$.

Problem 3 (Profile Threading with Non-strict Constraints): Given (s, A_s), (t, A_t), PF_t, and IC, find a threading (s', t') with the maximum score under the condition that

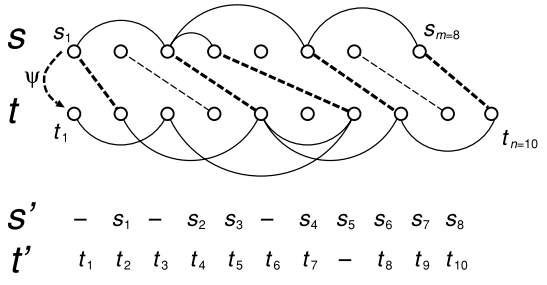

Fig. 1 Protein threading with constraints. For each arc in s, there must exist a corresponding arc in t which satisfies some constraints.

$\sum_{(s_i,s_{i'})\in A_s} IC(s_i, s_{i'}, \psi(s_i), \psi(s_{i'}))$ is the minimum.

It is worthy to notice that all the constraints must be satisfied in the former problem, whereas the latter problem tries to minimize the number of unsatisfied constraints.

3. Hardness Results

It is well-known that pairwise sequence alignment can be done in $O(mn)$ time using a simple dynamic programming algorithm [22]. It is also known that profile threading (Problem 1) can be done in $O(mn)$ time using a similar algorithm [22]. However, we show in this section that finding a feasible threading (i.e., a threading satisfying all the constraints) is NP-hard. Moreover, we show a strong hardness result on the approximation of an optimal feasible threading.

In this section, we only consider very simple constraints defined as: $IC(s_i, s_{i'}, t_j, t_{j'}) = 1$ iff. $(s_i, s_{i'}) \in A_s$ and $(t_j, t_{j'}) \notin A_t$. It should be noted that IC is uniquely determined from A_s and A_t. Similar constraints are used in the longest common subsequence problems with arc annotations [14]–[16].

Proposition 1: Both Problem 2 and Problem 3 are NP-hard.

Proof: We reduce the decision version of *maximum clique*, where maximum clique is a well-known NP-hard problem [23], [24]. Let $G(V, E)$ be an undirected graph where $V = \{v_1, \ldots, v_n\}$, and suppose that we are asked whether or not there exists a clique (i.e., a complete subgraph) $V' \subseteq V$ of size K.

From that instance, we construct a target sequence $s = s_1 \ldots s_K$ and a template structure $t = t_1 \ldots t_n$, where each s_i can be any amino acid. We define A_s, A_t by $A_s = \{(s_i, s_{i'}) |\ 1 \le i < i' \le K\}$ and $A_t = \{(t_j, t_{j'}) |\ 1 \le j < j' \le n$ and $\{v_j, v_{j'}\} \in E\}$, respectively. Then, it is easy to see that there exists a feasible threading if and only if there exists a clique of size K. □

In the above reduction, each s_i is connected with all other $s_{i'}$'s. Considering such protein structures is not realistic. However, we can still prove NP-hardness even if each s_i is connected with at most one residue (see Fig. 2).

Proposition 2: Both Problem 2 and Problem 3 are NP-hard even if each s_i appears at most once in A_s.

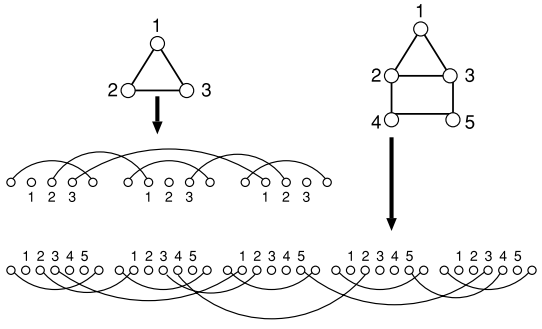

Fig. 2 Example of a reduction used in the proof of Proposition 2.

Proof: We modify the reduction used in the proof of Proposition 1 using a technique proposed in [7] and [13]. Let $G(V, E)$ be an instance of the maximum clique problem. We construct $s = s_1 s_2 \ldots s_{K(n+2)}$ and $t = t_1 t_2 \ldots t_{n(n+2)}$, where $n = |V|$, $s_{(h-1)(K+2)+1} s_{(h-1)(K+2)+2} \ldots s_{h(K+2))}$ corresponds to a vertex in a clique, and $t_{(k-1)(n+2)+1} t_{(k-1)(n+2)+2} \ldots t_{k(n+2))}$ corresponds to v_k. We define A_s and A_t by

$$\begin{aligned} A_s = {} & \{(s_{(h-1)(K+2)+1}, s_{h(K+2)}) |\ h = 1, \ldots, K\} \cup \\ & \{(s_{(h-1)(K+2)+1+k}, s_{(k-1)(K+2)+1+h}) | \\ & \qquad 1 \le h < k \le K\}, \\ A_t = {} & \{(t_{(h-1)(n+2)+1}, t_{h(n+2)}) |\ h = 1, \ldots, K\} \cup \\ & \{(t_{(h-1)(n+2)+1+k}, t_{(k-1)(n+2)+1+h}) | \\ & \qquad h < k, \{v_h, v_k\} \in E\}. \end{aligned}$$

Then, it is easy to see that there exists a feasible threading if and only if there exists a clique of size K. □

Next we study approximation hardness of Problem 2 (see [25] for terminologies on approximation algorithms). For that purpose, we assume that PF_t takes non-negative values. We also need the following lemma.

Lemma 1: Let $G(V, E)$ be an undirected graph where each vertex v_i has non-negative weight $w(v_i)$. The weight of a subgraph of G is defined to be the total weight of vertices in the subgraph. Let K be an integer. Then, the maximum weight K-clique of G can not be approximated within a factor of $O(|V|^{1-\epsilon})$ for any $\epsilon > 0$ unless NP=ZPP.

Proof: We show a *gap preserving reduction* [25] from the maximum clique problem to the maximum weight K-clique problem. Let $G'(V', E')$ be an instance of the maximum clique problem, where $n = |V'|$. We construct an instance of the maximum weight K-clique problem from $G'(V', E')$.

Let $G''(V'', E'')$ be a clique with n vertices. We construct $G(V, E)$ by $V = V' \cup V''$ and $E = E' \cup E'' \cup \{\{v_i, v_j\} \mid v_i \in V', v_j \in V''\}$, where $w(v_i) = 0$ if $v_i \in V''$, otherwise $w(v_i) = 1$.

Now we show that this is a gap preserving reduction. Suppose that $Q \subseteq V$ is an n-clique of $G(V, E)$ with weight W, where $K = n$ in this case. Then, $Q - V''$ will form a clique of $G'(V', E')$ with W vertices. On the other hand, suppose that $Q' \subseteq V'$ is a clique with W vertices. Then, we can obtain an n-clique of $G(V, E)$ with weight W by adding arbitrary $n - W$ vertices of V'' to Q'.

Since $|V| = 2n$ and maximum clique can not be approximated within a factor of $O(n^{1-\epsilon})$ for any $\epsilon > 0$ unless NP=ZPP [24], the lemma holds. □

Theorem 1: Problem 2 can not be approximated within a factor of $O(n^{1/2-\epsilon})$ for any $\epsilon > 0$ unless NP=ZPP.

Proof: We show a gap preserving reduction from the maximum weight K-clique problem.

Let $G(V, E)$ be an instance of the maximum weight K-clique problem constructed as in the proof of Lemma 1. From this graph and K, we construct an instance of Problem 2 as in the proof of Proposition 2. Moreover, we define PF_t by: $PF_t(s_i, t_{(j-1)(|V|+2)+1}) = 1$ for any s_i if $v_j \in V'$, otherwise

$PF_t(s_i, t_{j'}) = 0$.

Then, the score of an optimal feasible threading is equal to the weight of the maximum weight K-clique. Moreover, we can obtain a K-clique of weight W from a feasible threading with score W. Therefore, the theorem follows from [24] and $|t| = O(|V|^2)$. □

It should be noted that a simple score matrix between $|\Sigma| \times |\Sigma|$ (even for $|\Sigma| = 2$) can be used as PF_t in the above proof. It is also worthy to mention that Theorem 1 holds for Problem 3 if only a constant number of constraints are allowed to be violated.

4. Algorithms

4.1 CLIQUETHREAD

Although we used the maximum clique problem to show hardness results, it can also be used for solving protein threading with constraints. CLIQUETHREAD reduces the constrained threading problem to the maximum edge weight clique problem, in which the total weight for edges in the clique is maximized under the condition that the number of vertices of the clique is maximum. Though clique-based approach was also studied for structure alignment [26], existing methods can not be directly applied to our problem because affine gap costs and profiles are considered in this paper whereas they solved the structure alignment problem in discrete settings. Though an improved version of CLIQUETHREAD was developed in the companion paper [18], we show here the original version of CLIQUETHREAD.

We construct an instance $G(V, E)$ of the clique problem in the following way. Let $s_{i_1}, s_{i_2}, \ldots, s_{i_H}$ be residues in s appearing in A_s, where $i_1 < i_2 < \ldots < i_H$. We construct an undirected graph $G(V, E)$ defined by

$$
\begin{aligned}
V &= \{(s_{i_h}, t_j) | 1 \le h \le H, 1 \le j \le n\} \cup \{v_0, v_e\}, \\
E &= \{\{(s_{i_h}, t_j), (s_{i_{h'}}, t_{j'})\} \mid 1 \le h < h' \le H, \\
&\qquad 1 \le j < j' < n\} \cup \\
&\qquad \{\{v_0, (s_{i_h}, t_j)\} \mid 1 \le h \le H, 1 \le j \le n\} \cup \\
&\qquad \{\{(s_{i_h}, t_j), v_e\} \mid 1 \le h \le H, 1 \le j \le n\}.
\end{aligned}
$$

Then, the weight of each edge is given by equations as in Sect. 3.2 of [18]. The only difference is that if the distance constraints are satisfied then irrespective of the score of the alignment, the corresponding weight is assigned to the edge here, whereas in Sect. 3.2 of [18], even if the distance constraints are satisfied, the weight assigned is 0 if the score of the alignment is less than some threshold value.

For this graph, the size of the maximum cardinality clique is $H + 2$ if there exists a feasible threading. Moreover, the maximum cardinality clique consists of vertices of the form:

$$v_0, \ (s_{i_1}, t_{j_1}), \ (s_{i_2}, t_{j_2}), \ \ldots, \ (s_{i_H}, t_{j_H}), \ v_e.$$

Each of the maximum cardinality cliques corresponds to a threading in which $\psi(s_{i_h}) = t_{j_h}$ holds for all $h = 1, 2, \ldots, H$. Then, the score of an optimal feasible threading is given by $W - \alpha(H + 2)(H + 1)/2$, where W is the total weight of the maximum edge weight clique, and we assume that a constant α (see [18] for detals) is much larger than the possible threading scores.

Therefore, we can obtain a solution for Problem 2 by solving the maximum edge weight clique problem. Even if all the constraints are not satisfied, CLIQUETHREAD tries to minimize the number of s_{i_k}'s violating constraints though there is no theoretical guarantee on the scores of computed threadings.

Now we analyze the time complexity of the reduction procedure. The number of vertices of $G(V, E)$ is clearly $nH+2$. The number of edges is $O((nH)^2)$. For $O(H \cdot n^2)$ pairs of substrings $(s_{i_k}, \ldots, s_{i_{k+1}}, t_j \ldots t'_j)$, we compute the score of an optimal threading without constraint. It would take $O(Hmn^3)$ time in total. However, we can use the same DP matrix for computing the scores for $(s_{i_k}, \ldots, s_{i_{k+1}}, t_j \ldots t_{j+1})$, $(s_{i_k} \ldots s_{i_{k+1}}, t_j \ldots t_{j+2})$, $(s_{i_k} \ldots s_{i_{k+1}}, t_j \ldots t_{j+3})$, $\cdots$. Moreover, $\sum_{i_k} |s_{i_k} \ldots s_{i_{k+1}}| = O(m)$. Thus, the total time for computing the scores of optimal threadings for substring pairs is $O(mn^2)$. Therefore, we have:

Theorem 2: Problem 2 can be reduced to the maximum edge weight clique problem with $|V| = nH + 2$ in $O((H^2 + m)n^2)$ time.

Here, we briefly discuss about practical computation time. Usually, n is at most a few thousands and H is at most a few tens. If we use the experimental method proposed by [5], we can only measure distances between Lys residues. In such a case, $H \approx m/20$. Thus, the time for reduction is not so crucial. Furthermore, the number of vertices of $G(V, E)$ is usually at most several thousands. Experimental results shown in [9], [10] suggest that the fastest (maximum edge weight) clique algorithms can solve instances of size ($|V|$) 1000 $\sim$ 10000. Thus, the proposed method is practical for non-large proteins (for proteins with less than 200 residues in our experiments). Furthermore, CLIQUETHREAD has another potential merit: all practical instances may be solved if a much faster clique algorithm is developed in the future.

4.2 BBDPTHREAD

We showed in our previous work [7] that protein threading with pairwise energy function can be solved exactly in polynomial time using a DP algorithm if the graph representing interactions between core regions has a tree-like (nested) structure. Jiang et al. developed similar algorithms for special cases of the longest common subsequence problem for arc annotated sequences [14], [15]. In this section, we develop a similar DP algorithm for a special case of Problem 2 and then combine it with a branch-and-bound procedure for a general case.

We consider a special case in which any two pairs in A_s do not cross (i.e., there are no pairs $(s_i, s_{i'})$, $(s_k, s_{k'})$ in A_s with $i < k < i' < k'$ or $k < i < k' < i'$). In that case, we have

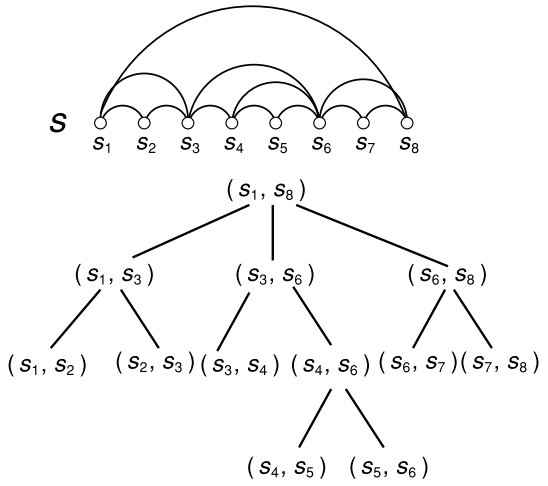

Fig. 3 Example of a tree structure used in BBDPTHREAD.

a tree structure in the following way.

For each pair $(s_{i_k}, s_{i_{k'}})$ such that $(s_{i_k}, s_{i_{k'}}) \in A_s$, $(s_{i_h}, s_{i_{h'}}) \in A_s$ is called an *ancestor* if either $i_h \le i_k < i_{k'} < i_{h'}$ or $i_h < i_k < i_{k'} \le i_{h'}$. Moreover, $(s_{i_h}, s_{i_{h'}})$ is called a *parent* of $(s_{i_k}, s_{i_{k'}})$ if there exists no $(s_{i_g}, s_{i_{g'}})$ such that $(s_{i_g}, s_{i_{g'}})$ is an ancestor of $(s_{i_k}, s_{i_{k'}})$ and $(s_{i_h}, s_{i_{h'}})$ is an ancestor of $(s_{i_g}, s_{i_{g'}})$. Then, a tree structure is induced by this relationship (see Fig. 3). We assume without loss of generality that $(s_1, s_m) \in A_s$, $IC(s_1, s_m, t_1, t_n) = 0$ and $(s_{i_k}, s_{i_{k+1}}) \in A_s$ for all i_k.

We compute score $S(s_{i_k}, s_{i_{k'}}, t_j, t_{j'})$ using DP, where $S(s_{i_k}, s_{i_{k'}}, t_j, t_{j'})$ denotes the maximum score of all feasible threadings between $s_{i_k} \ldots s_{i_{k'}-1}$ and $t_j \ldots t_{j'-1}$ under the condition that $\psi(s_{i_k}) = t_j$ and $\psi(s_{i_{k'}}) = t_{j'}$. If there is no feasible threading or $IC(s_{i_k}, s_{i_{k'}-1}, t_j, t_{j'-1}) = 1$, we let $S(s_{i_k}, s_{i'_k}, t_j, t_{j'}) = -\infty$. Clearly, $S(s_1, s_m, t_1, t_n)$ denotes the score of an optimal feasible threading between s and t. If there is no feasible threading, $S(s_1, s_m, t_1, t_n) = -\infty$.

We compute $S(s_{i_k}, s_{i'_k}, t_j, t_{j'})$ in a bottom up manner (i.e., from leaves to the root). For each leaf $(s_{i_k}, s_{i_{k+1}})$, we compute the score by

$$S(s_{i_k}, s_{i_{k+1}}, t_j, t_{j'}) = \begin{cases} -\infty & \text{if } IC(s_{i_k}, s_{i_{k+1}}, t_j, t_{j'}) = 1, \\ score(s_{i_k+1} \ldots s_{i_{k+1}-1}, t_{j+1} \ldots t_{j'-1}) + \\ \qquad PF_t(s_{i_k}, t_j) & \text{otherwise,} \end{cases}$$

where $score(s'', t'')$ denotes the score of an optimal threading without constraints (i.e., an optimal solution for Problem 1) between substrings s'' and t'', and the expression in the last line should be replaced by $score(s_{i_k+1} s_{i_k+2} \ldots s_{i_{k+1}-1}, t_{j+1} t_{j+2} \ldots t_{j'-1}) + PF_t(s_{i_k}, t_j) + PF_t(s_{i_{k+1}}, t_{j'})$ in the case of $i_{k+1} = n$.

For each non-leaf node $(s_{i_k}, s_{i_{k'}})$, let $(s_{i_k} = s_{r_1}, s_{r_2}, \ldots, s_{r_p} = s_{i_{k'}})$ be a sequence of residues in s such that $(s_{i_k}, s_{i_{k'}})$ is a parent of $(s_{r_q}, s_{r_{q+1}})$, where $r_1 < r_2 < \cdots < r_p$. For example, (s_1, s_3, s_6, s_8) is such a sequence for (s_1, s_8) in Fig. 3. Here, we can assume that values of $S(s_{r_q}, s_{r_{q+1}}, t_h, t_{h'})$'s are already computed before computing $S(s_{i_k}, s_{i_{k+1}}, t_j, t_{j'})$. Then, we compute $S'(r_q, j, j')$ by the following dynamic programming procedure:

$$S'(r_2, j, j') = S(s_{r_1}, s_{r_2}, t_j, t_{j'}),$$
$$S'(r_{q+1}, j, j') = \max_{j < j'' < j'} \{S'(r_q, j, j'') + S(s_{r_q}, s_{r_{q+1}}, t_{j''}, t_{j'})\}.$$

Then, we let

$$S(s_{i_k}, s_{i_{k+1}}, t_j, t_{j'}) = \begin{cases} -\infty & \text{if } IC(s_{i_k}, s_{i_{k+1}}, t_j, t_{j'}) = 1, \\ S'(i_{k+1}, j, j') & \text{otherwise.} \end{cases}$$

Finally, $S(s_1, s_m, t_1, t_n)$ gives the score of an optimal feasible threading. Here we briefly analyze the time complexity. In the dynamic programming procedure, it takes $O(n)$ time per $S'(r_{q+1}, j, j')$. Since there exist $O(H \cdot n^2)$ combinations of $S'(r_{q+1}, j, j')$, it takes $O(Hn^3)$ time in total. As in CLIQUETHREAD, the total time required for computing the scores of optimal threadings for substrings is $O(mn^2)$.

Theorem 3: If any two pairs in A_s do not cross, Problem 2 can be solved in $O(Hn^3 + mn^2)$ time.

In most practical cases, the above condition is not satisfied. However, the condition is satisfied in many practical cases if we remove several s_i's from A_s. This fact leads to a combination of exhaustive search and the DP algorithm.

Suppose that A'_s is obtained by removing the minimum number of residues $(s_{u_1}, \ldots, s_{u_D})$ from A_s so that the condition of Theorem 3 is satisfied for A'_s. For all combinations of $t_{b_1}, \ldots, t_{b_D}$ (which are candidates of $\psi(s_{u_1}), \ldots, \psi(s_{u_D})$), we apply a modified DP procedure for A'_s without removing $s_{u_1}, \ldots, s_{u_D}$ from $s_{i_1}, \ldots, s_{i_H}$. In the modified DP procedure, we check if the constraints relevant to $s_{u_1}, \ldots, s_{u_D}$ are satisfied. For that purpose, we compute a table ICs_{i_k} for each s_{i_k}. $ICs_{i_k}(j) = 1$ if and only if $IC(s_{i_k}, s_{u_h}, t_j, t_{b_h}) = 1$ or $IC(s_{u_h}, s_{i_k}, t_{b_h}, t_j) = 1$ holds for some s_{u_h}. Then, we can check constraints relevant to A'_s at the computation of the score for each leaf in constant time. If the condition is not satisfied, we let the score for the leaf to be $-\infty$. Clearly, this exhaustive procedure takes $O(n^D \cdot (Hn^3 + mn^2))$ time.

BBDPTHREAD uses a simple branch-and-bound procedure in order to reduce the practical computation time. The pseudocode of BBDPTHREAD is given below, where L corresponds to $t_{b_1}, \ldots, t_{b_d}$, and $L \cdot x$ means that x is appended to L.

```
Procedure BBDPTHREAD(s_{u_1}, ..., s_{u_D})
  for all s_{u_d} do
    for all j do T_d(j) ← the score of an optimal
        threading under the constraints relevant
        to A'_s and ψ(s_{u_d}) = j
    sort T_d in the decreasing order
  S_max ← −∞;  BBDP({})

Procedure BBDP(L)
  if (|L| = D) then
```

compute the score S_L of an optimal threading for L
if $S_L > S_{\max}$ **then** $S_{\max} \leftarrow S_L$
return
if ($|L| > 0$) **then**
compute the score S_L of an optimal threading for L
if $S_L < S_{\max}$ **then return**
for all j in decreasing order of $T_{|L|+1}(j)$ **do**
if $T_{|L|+1}(j) < S_{\max}$ **then return**
$BBDP(L \cdot t_j)$

In the practical version, we execute $BBDP(\{\})$ for several appropriate initial values of $S_{\max}$ instead of $S_{\max} = -\infty$.

We currently use exhaustive search for selecting $s_{u_1}, \ldots, s_{u_D}$ since BBDPTHREAD does not work if D is large (e.g., ≥ 10). Instead, we may use the following greedy procedure: select s_{u_i} with the maximum crossing edges at i-th greedy step.

5. Computational Experiments

We performed computational experiments on CLIQUETHREAD and BBDPTHREAD in order to evaluate practical computation time and usefulness for improving the accuracy of profile threading. As target and template protein pairs, we tested structure data of 9 protein pairs in PDB [11], which belong to major fold classes of all α proteins, all β proteins and α-and-β proteins. We used a PC cluster with Intel Xeon 2.8 GHz CPUs, where it was working under the LINUX operating system. Though we used a PC cluster, each algorithm was executed using only one CPU. All algorithms were implemented using C language. The distance constraints and the parameters used are same as those used in our related work [18].

For CLIQUETHREAD, we employed the maximum edge-weight clique algorithm developed by the authors, which was shown to be one of the fastest clique algorithms using DIMACS benchmark data [9], [10].

For the comparison of CPU times of CLIQUETHREAD and BBDPTHREAD, please refer to Table 1 of the companion paper [18]. The table shows that in the case of protein pairs 1bbn/1cnt1, 1xyzA/8timA and 1atnA/1atr, the time taken by CLIQUETHREAD/BBDPTHREAD are 1.5 s/8.3 s, 3279 s/59.9 s and NA/1101 s respectively. NA means the execution did not finish with in 10 hours.

From this result, it can be observed that CLIQUETHREAD is faster than BBDPTHREAD for smaller proteins, while BBDPTHREAD shows better performance than CLIQUETHREAD for large proteins (e.g., up to proteins with 300-400 residues). However, CLIQUETHREAD has some merits: (i) CLIQUETHREAD is faster when distances between all Lys-Lys pairs are given (because the clique algorithm works efficiently in this case), (ii) CLIQUETHREAD can output a reasonable alignment even if there does not exist a feasible threading. For example, we examined the case of 1atnA/1atr pair in which all Lys-Lys distances (i.e., including distances > 24 Å) were used as constraints. In this case, there did not exist a feasible threading and thus BBDPTHREAD failed to output an alignment. However, CLIQUETHREAD output a threading within 31 seconds, in which 300 residue pairs are superimposed with RMSD= 11.14 Å (recall that CLIQUETHREAD took more than 10 hours when Lys-Lys pairs of distances at most 24 Å were used as constraints). Therefore, CLIQUETHREAD might be useful when Lys-Lys distances more than 24 Å are given and/or there does not exist a feasible threading.

Table 1 Comparison of threading results. The second column shows the results without using constraints, the third column shows the results with constraints and the fourth column shows the results of STRALIGN. In each column, the RMSD and the number of aligned residue pairs are shown in the form of x/y where x denotes the RMSD and y denotes the number of aligned residues.

Target/ Template	No Constraints	With Constraints	Structure Alignment
1bbn/1cnt1	10.35 / 119	8.97 / 127	2.09 / 96
1vltA/1nfn	16.37 / 119	11.43 / 103	2.02 / 102
3sdhA/1dlw	9.10 / 112	6.19 / 106	2.17 / 102
1ten/1ac6A	14.52 / 81	14.04 / 80	2.20 / 70
1bla/1hce	15.06 / 106	4.12 / 118	1.93 / 108
1a3k/1f5f	14.73 / 130	9.14 / 130	2.18 / 106
1bow/1d5yA2	15.39 / 137	14.17 / 135	2.03 / 104
1xyzA/8timA	16.68 / 214	13.15 / 197	2.65 / 130
1atnA/1atr	14.41 / 314	9.01 / 299	1.97 / 260

The performance comparison of our constrained threading algorithm is assessed by comparing our algorithm with the unconstrained threading using the same scoring function (sequence-profile score). This is done because there does not exist any constrained threading method that directly uses the profiles of PSI-BLAST.

The accuracies of obtained threadings are summarized in Table 1, where RMSD (*Root Mean Square Deviation*, Å) between the superimposed C^α atoms and the number of superimposed residues (i.e., the number of aligned residue pairs) are shown for each case. RMSD is the scoring function to indicated how closely fit are the two aligned structures.

Let us consider x_i and y_i be the corresponding C^α atoms of two structures whose RMSD has to be calculated and N be the number of C^α residues then the RMSD is given by $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2}$. The lesser is the RMSD, the better the similarity between the two compared structures. We also listed the results of structural alignment for the evaluation. We employed STRALIGN (http://www.hgc.ims.u-tokyo.ac.jp/service/tooldoc/stralign) [27] for structure alignment, where it was developed by the authors and its performance was considered to be comparable to other structure alignment algorithms. It should be noted that complete structural data of two input proteins are given in structure alignment, while structural data of one input protein is given in protein threading. The results of structure alignment can be considered as almost the *correct answers*.

As seen from Table 1, it can be concluded that the accuracies obtained by constrained threading are in general better than those by unconstrained threading which justifies that constraints help in obtaining better results. Moreover, it should be noted that much better results were obtained by constrained threading for 1bla/1hce and 1atnA/1atr pairs and hence it can be observed that the constraints are useful in increasing the efficiency of the prediction. Compared with the results of structural alignment, the results of constrained threading were not good for 1ten/1ac6A, 1bow/1d5yA2 and 1xyzA/8timA pairs. However, it is reasonable because the numbers of superimposed residues by structural alignment are small in these cases, compared with the sizes of input structures.

6. Concluding Remarks

In this paper, we showed that protein threading with profiles and constraints is very hard from a theoretical viewpoint. This result also suggests that protein threading with pairwise energy and constraints is very hard because threading with pairwise energy is much harder than threading with profiles [7].

From a practical viewpoint, it was shown that information about Lys-Lys distances is useful to improve the alignment accuracy of profile threading. It was also shown that the proposed algorithms (especially, BBDPTHREAD) are useful for threading with up to medium size protein structures. However, these take very long time for large protein structures. Therefore, improvement of efficiency of the algorithms is important future work. In particular, there is much room for improvement on BBDPTHREAD because a simple branch-and-bound procedure is employed in the current version. More rigorous computational experiments, especially experiments on fold recognition, are important future work, too. Though we considered the threading approach for structure prediction with constraints, the *ab-initio* approach should also be studied [28].

It is interesting that Propositions 1 and 2, and the DP algorithm used in BBDPTHREAD are similar to our previous results on protein threading with pairwise energy [7] though the roles of a target sequence and a template structure were exchanged there: arcs for target sequences were mainly considered in BBDPTHREAD, while arcs for template sequences were considered in [7]. Using this property and treating each residue in A_s as a core region, most exact algorithms for protein threading with pairwise energy [1], [19], [20] might be modified for protein threading with profiles and constraints in which gaps are allowed everywhere (note that gaps are not allowed in core regions if simple modifications are done for the existing algorithms). Such modifications might be useful for developing faster algorithms for protein threading with profiles and constraints. Conversely, it would also be interesting to apply the techniques used in CLIQUETHREAD and BBDPTHREAD to other related problems such as RNA structure comparison.

Acknowledgements

This work was supported in part by Grants-in-Aid #17017019 and #16300092 from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan, and by the Kayamori Foundation of Information Science Advancement.

References

[1] Y. Xu, D. Xu, O.H. Crawford, and J.R. Einstein, "A computational method for NMR-constrained protein threading," J. Comput. Biol., vol.7, pp.449–467, 2000.
[2] M. Albrecht, D. Hanisch, R. Zimmer, and T. Lengauer, "Improving fold recognition of protein threading by experimental distance constraints," Insilico Biol., vol.2, no.3, pp.325–337, 2002.
[3] J. Meiler and D. Baker, "Rapid protein fold determination using unassigned NMR data," Proc. Natl. Acad. Sci. USA, vol.100, pp.15404–15409, 2003.
[4] W. Li, Y. Zhang. D. Kihara, Y.J. Huang, D. Zheng, G.T., Montelion, A. Kolinski, and J. Skolnick, "TOUCHSTONEX: Protein structure prediction with sparse NMR data," Proteins: Struct. Funct. Genet., vol.53, pp.290–306, 2003.
[5] M.M. Young, N. Tang, J.C. Hempel, C.M. Oshiro, E.W. Taylor, I.D. Kuntz, B.W. Gibson, and G. Dollinger, "High throughput protein fold identification by using experimental constraints derived from intermolecular cross-links and mass spectrometry," Proc. Natl. Acad. Sci. USA, vol.97, pp.5802–5806, 2000.
[6] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," Nucleic Acids Res., vol.25, pp.3389–3402, 1997.
[7] T. Akutsu and S. Miyano, "On the approximation of protein threading," Theor. Comput. Sci., vol.210, pp.261–275, 1999. (also in Proc. RECOMB 1997, pp.3–8).
[8] R.H. Lathrop, "The protein threading problem with sequence amino acid interaction preferences is NP-complete," Protein Eng., vol.7, pp.1059–1068, 1994.
[9] J. Suzuki, E. Tomita, and T. Seki, "An algorithm for finding a maximum clique with maximum edge-weight and computational experiments," Technical Report MPS-42-12, pp.45–48, Information Processing Society of Japan, 2002.
[10] E. Tomita and T. Seki, "An efficient branch-and-bound algorithm for finding a maximum clique," Lect. Notes Comput. Sci., no.2731, (Proc. DMTCS 2003), pp.278–289, 2003.
[11] H.M. Berman, J.D. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne, "The protein data bank," Nucleic Acids Res., vol.28, pp.235–242, 2000.
[12] D. Goldman, S. Istrail, and C.H. Papadimitriou, "Algorithmic aspects of protein structure similarity," Proc. 40th IEEE Symp. on Foundations of Computer Science, pp.512–522, 1999.
[13] P.A. Evans, "Finding common subsequences with arcs and pseudoknots," Lect. Notes Comput. Sci., no.1645 (Proc. CPM 1999), pp.270–280, 1999.
[14] T. Jiang, G. Lin, B. Ma, and K. Zhang, "The longest common subsequence problem for with arc annotated sequences," Lect. Notes Comput. Sci., no.1848 (Proc. CPM 2000), pp.154–165, 2000.
[15] T. Jiang, G. Lin, B. Ma, and K. Zhang, "A general edit distance between RNA structures," J. Comput. Biol., vol.9, pp.371–388, 2002.
[16] G. Lin, Z.-Z. Chen, T. Jiang, and J. Wen, "The longest common subsequence problem for sequences with nested arc annotations," J. Comput. Syst. Sci., vol.65, pp.465–480, 2002.
[17] D. Bahadur K.C., E. Tomita, J. Suzuki, K. Horimoto, and T. Akutsu, "Clique based algorithms for protein threading with profiles

and constraints," Proc. 3rd Asia Pacific Bioinformatics Conference (APBC2005), pp.51–64, Singapore, 2005.
[18] D. Bahadur K.C., E. Tomita, J. Suzuki, K. Horimoto, and T. Akutsu, "Protein threading with profiles and distance constraints using clique based algorithms," J. Bioinformatics and Computational Biology, in press.
[19] J. Xu, M. Li, D. Kim, and Y. Xu, "RAPTOR: Optimal protein threading by linear programming," J. Bioinformatics and Computational Biology, vol.1, pp.95–118, 2003.
[20] R.H. Lathrop and T.F. Smith, "Global optimum protein threading with gapped alignment and empirical pair score functions," J. Mol. Biol., vol.255, pp.641–665,1996.
[21] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider, "The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003," Nucleic Acids Res., vol.31, pp.365–370, 2003.
[22] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, Biological Sequence Analysis, Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, Cambridge, UK, 1998.
[23] M.R. Garey and D.S. Johnson, Computers and Intractability, Freeman, New York, 1979.
[24] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$," Proc. 37th IEEE Symp. on Foundations of Computer Science, pp.627–636, 1996.
[25] V.V. Vazirani, Approximation Algorithms, Springer, Berlin, 2001.
[26] G. Lancia, R. Carr, B. Walenz, and S. Istrail, "101 optimal PDB structure alignments: A branch-and-cut algorithm for the maximum contact map overlap problem," Proc. 5th Int. Conf. Computational Molecular Biology, pp.193–202, Canada, 2001.
[27] T. Akutsu, "Protein structure alignment using dynamic programming and iterative improvement," IEICE Trans. Inf. & Syst., vol.E79-D, no.12, pp.1629–1636, Dec. 1996.
[28] K. Yue and K.A. Dill, "Block constraint-based assembly of tertiary protein structures from secondary structure elements," Protein Sci., vol.9, pp.1935–1946, 2000.

Tatsuya Akutsu received his M.Eng degree in Aeronautics in 1996 and a Dr. Eng. degree in Information Engineering in 1989 both from University of Tokyo, Japan. From 1989 to 1994, he was with Mechanical Engineering Laboratory, Japan. He was an associate professor in Gunma University from 1994 to 1996 and in Human Genome Center, University of Tokyo from 1996 to 2001 respectively. He joined Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan as a professor in Oct. 2001. His research interests include bioinformatics and discrete algorithms.

Morihiro Hayashida is currently a research associate in the Laboratory of Biological Information Networks, Bioinformatics Center at Kyoto University. He received his Masters degree from Graduate School of Information Science at The University of Tokyo and Doctors degree from Kyoto University. His research interests include functional analysis of proteins and development of computational methods for protein interaction prediction.

Dukka Bahadur K.C. is currently a Ph.D. course student in Laboratory of Biological Network analysis, Bioinformatics Center, Kyoto University. He received his B.Eng. in 2001 and M.Inf. degree in 2003 both from Kyoto University. His current research interests include development of computational methods for protein structure prediction.

Etsuji Tomita received his B.Eng. and Dr.Eng. degrees in Electronics Engineering from Tokyo Institute of Technology, Japan, in 1966 and 1971, respectively. Then he was with the faculties of Tokyo Tech., and was appointed Associate Professor at the University of Electro-Commmunications, Japan. Since 1986, he has been a Professor at UEC. His research interests include combinatorial optimization problems, theory of automata and formal languages, and algorithmic learning theory. He was awarded the Funai Information Technology Prize in 2003, and is presently a Fellow of IPSJ.

Jun'ichi Suzuki graduated from the Department of Information and Communication Engineering, the University of Electro-Communications, Japan, in March 2003. Since April 2003, he has been in the Graduate School of Electro-Communications, the University of Electro-Communications, Japan. He was given the IPSJ Yamashita SIG Research Award in July 2003. He has been working in the algorithms for solving combinatorial optimization problems.

Katsuhisa Horimoto received his M.Sc. degree in Biophysics from Science University of Tokyo, Japan, and his Ph.D. degree in Biophysics from Science University of Tokyo, Japan, in 1991. From April 1991 to March 1997, he was at Science University of Tokyo as a research associate, and from April 1997 to September 2001, he was at Saga Medical School as an associate professor. Since October 2001, he has been a professor in Institute of Medical Science, University of Tokyo.