Original Paper

Hardness Results on Local Multiple Alignment of Biological Sequences

Tatsuya Akutsu,† Hiroki Arimura†† and Shinichi Shimozono†††

This paper studies the local multiple alignment problem, which is, given protein or DNA sequences, to locate a region (i.e., a substring) of fixed length from each sequence so that the score determined from the set of regions is optimized. We consider the following scoring schemes: the relative entropy score (i.e., average information content), the sum-of-pairs score and a relative entropy-like score introduced by Li, et al. We prove that multiple local alignment is NP-hard under each of these scoring schemes. In particular, we prove that multiple local alignment is APX-hard under relative entropy scoring. It implies that unless P = NP there is no polynomial time algorithm whose worst case approximation error can be arbitrarily specified (precisely, a polynomial time approximation scheme). Several related theoretical results are also provided.

1. Introduction

Multiple sequence alignment is one of the well studied problems in computational molecular biology and has many applications. For example, it is useful for locating binding sites, finding conserved regions, and building phylogenetic trees [6),17),18)]. This problem is divided into *global multiple alignment* and *local multiple alignment* [14)]. The goal of global multiple alignment is to align complete sequences, whereas the aim of local multiple alignment is to locate relatively short patterns shared by sequences. This paper focuses on local multiple alignment [9),10),13)~15),17),18)]. Local multiple alignment is useful for finding binding sites, conserved regions and motifs of sequences.

Local multiple alignment is a problem of, given n sequences, locating a region (i.e., a substring) of fixed length from each sequence so that the *score* determined from the set of regions is optimized. So far, several scoring schemes have been proposed. Local multiple alignment is also known as the *global consensus patterns* problem [15)].

Many studies have been done on local multiple alignment. Stormo and Hartzell proposed the score based on relative entropy (average information content) and developed a heuristic iterative algorithm for finding an optimal score [17),18)]. Since this scoring scheme is based on an appropriate statistical model of biological sequences, it has been widely used in practice along with variants [6),9),10),13),14)]. Under this scoring scheme, Lawrence and Reilly developed an EM (*expectation maximization*) algorithm [13)], Lawrence, et al. developed a Gibbs sampling algorithm [14)], and Horton developed branch-and-bound algorithms [9),10)]. However, these algorithms except Horton's algorithms are not guaranteed to find an optimal alignment (i.e., an alignment with the maximum score). Any theoretical guarantee is not given for the scores of the computed alignments. Although Horton's algorithms always find optimal alignments, they are not efficient (i.e., they are not polynomial time algorithms). Li, Ma and Wang developed a polynomial time approximation algorithm (we call it the LMW algorithm) for local multiple alignment under relative entropy scoring along with algorithms for some other scoring schemes [15)]. The most important feature of the algorithm is that it has a theoretical guarantee on the error (the difference between the optimal score and the score of the computed alignment). And also, the algorithm was proven to be a polynomial time approximation algorithm whose worst case approximation error can be arbitrarily specified as an auxiliary parameter (precisely, a polynomial time approximation scheme) under a scoring scheme called the #LOG#-scoring in our paper. However, the running time of the algorithm depends exponentially on that parameter, and so in practice a huge amount of time is needed to keep the approximation error to be small.

† Bioinformatics Center, Institute for Chemical Research, Kyoto University
†† Graduate School of Information Science and Technology, Hokkaido University
††† Department of Artificial Intelligence, Kyushu Institute of Technology

In this paper, we consider local multiple alignment under the following scoring schemes: the *relative entropy score* [13),14),17),18)], the #LOG#-score introduced in Ref. 15), and the SP-score (the *sum-of-pairs score*) [5),8),20)]. Though SP-score has not been used for local multiple alignment in practice, a lot of theoretical and practical studies have been done on global multiple alignment under SP-scoring and it variants (e.g., the weighted sum-of-pairs scoring) [6),8),19)]. Thus, it is interesting to study local multiple alignment under SP-scoring at least from a theoretical viewpoint though it is not relevant from a practical viewpoint. We prove that local multiple alignment is NP-hard under each of these scoring schemes. In particular, we prove that local multiple alignment under relative entropy scoring is APX-hard, which implies that no *polynomial time approximation scheme* (PTAS) exists unless P = NP [2),4)]. Although NP-hardness results were proven for global multiple alignment under SP-scoring [20)] and related problems [15)], to our knowledge, there had been no known non-approximability results on local multiple alignment under relative entropy scoring ☆.

Also, we have developed a new, extremely simple PTAS under #LOG#-scoring, though the LMW algorithm is conceptually simple. Compared with this PTAS, we have made an observation that #LOG#-scoring is by no means adequate for evaluating the quality of local multiple alignment. Furthermore, we show that a technique used in an approximation algorithm for global multiple alignment under SP-scoring [8)] can also be used for designing an approximation algorithm for local multiple alignment under SP-scoring.

2. Problem and Scoring Schemes

In this section, we define the local multiple alignment problem and the scoring schemes formally. We use notations similar to those in Ref. 15).

Let Σ be an alphabet of size A. Usually, $\Sigma = \{\mathrm{A,C,G,T}\}$ or Σ consists of letters denoting amino acid residues (i.e., $A = 4$ or $A = 20$). For a string s over Σ, $|s|$ denotes the length of s. $s[i]$ is the i-th character of s. Thus, $s = s[1]s[2]\ldots s[|s|]$. We define the local multiple alignment problem as follows (see also **Fig. 1**).

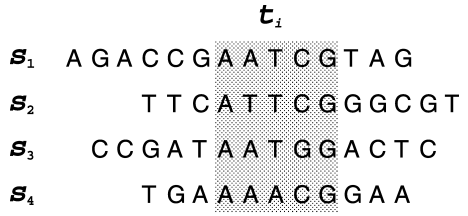

Fig. 1 Example of local multiple alignment. In this case, $f_1(\mathrm{A}) = 1.0$, $f_2(\mathrm{A}) = 0.75$, $f_2(\mathrm{T}) = 0.25$, $f_3(\mathrm{A}) = 0.25$, $f_3(\mathrm{T}) = 0.75$, $f_4(\mathrm{C}) = 0.75$, $f_4(\mathrm{G}) = 0.25$, $f_5(\mathrm{G}) = 1.0$, and $f_j(a) = 0.0$ for other a, j.

Local Multiple Alignment: Given a set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of sequences, and an integer L, find a substring t_i of length L from each s_i, maximizing the score of $(t_1, \cdots, t_n)$. We call $(t_1, \ldots, t_n)$ a local multiple alignment, a local alignment, or simply an alignment.

Although each input string must be of the same length in Ref. 15), strings with different lengths are input in practice and thus we employ this definition.

Let $\#_j(a)$ be the number of the appearances of letter a in the j-th column of t_i's (i.e., $\#_j(a) = |\{t_i | t_i[j] = a\}|$). Let $f_j(a)$ be the frequency of letter a in the j-th column of t_i's (i.e., $f_j(a) = \frac{\#_j(a)}{n}$). Let $p(a)$ denote the frequency of letter a in the whole genome (i.e., background probability of a). We consider the following three scoring schemes ☆☆.

#LOG#-score: [15)]

$$score(t_1, \ldots, t_n) = \sum_{j=1}^{L} \sum_{a \in \Sigma} \#_j(a) \log \#_j(a),$$

Relative entropy score: (*average information content*) [9),10),13)~15),17),18)]

$$score(t_1, \ldots, t_n) = \frac{1}{L} \sum_{j=1}^{L} \sum_{a \in \Sigma} f_j(a) \log \frac{f_j(a)}{p(a)},$$

SP-score: (*sum-of-pairs*) [5),8),20)]

$$score(t_1, \ldots, t_n) = \sum_{j=1}^{L} \sum_{i<i'} dist(t_i[j], t_{i'}[j]),$$

where $dist(x, y)$ is the distance between letter x and letter y. As in Refs. 5), 8), 20), we consider an arbitrary distance satisfying the triangle inequality and thus the problem in this case is defined as the *minimization problem* instead of the maximization problem.

Given an instance $\mathcal{I}$ of the problem, $OPT(\mathcal{I})$ denotes the score of an optimal solution of

☆ Preliminary results were included in our previous conference paper [1)].

☆☆ In this paper, $\log x$ means $\log_2 x$ and we define $0 \log 0 \equiv 0$.

$\mathcal{I}$. For a maximization problem, an algorithm $\mathcal{A}$ is called a PTAS if, for any instance $\mathcal{I}$ of the problem and for any *constant* $0 < \epsilon < 1$, $\mathcal{A}$ always outputs a solution $\mathcal{X}$ satisfying $score(\mathcal{X}) \geq (1-\epsilon) \cdot OPT(\mathcal{I})$ in polynomial time (for a minimization problem, we replace $score(\mathcal{X}) \geq (1-\epsilon) \cdot OPT(\mathcal{I})$ with $score(\mathcal{X}) \leq (1+\epsilon) \cdot OPT(\mathcal{I})$) [4].

3. Results on #LOG#-score

Li, Ma and Wang dealt with Local Multiple Alignment under #LOG#-scoring over a fixed alphabet [15]. Here, we show that Local Multiple Alignment under #LOG#-scoring is APX-hard if an alphabet Σ is unbounded. This implies that if we allow arbitrarily many kind of symbols in inputs then there is no PTAS even under #LOG#-scoring. The proof is not difficult, but the technique employed here will be also applied to prove the APX-hardness for relative entropy score.

Theorem 3.1 Local Multiple Alignment under #LOG#-scoring is APX-hard if an alphabet Σ is unbounded.

Proof. We show an L-reduction [16] from Max Cut. Recall that Max Cut is, given an undirected graph $G(V,E)$, to find a partition (V_1, V_2) of V (i.e., $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$) maximizing the number of edges between V_1 and V_2. It is known that Max Cut is APX-hard [3),16)].

Let $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. From this instance, we construct n sequences $s_1, \ldots, s_n$ each of length $3m$. For each edge $e_k = \{v_i, v_j\} \in E$ $(i < j)$, we let

$$s_i[k] = a_k, \quad s_i[2m+k] = b_k,$$
$$s_j[k] = b_k, \quad s_j[2m+k] = a_k,$$

where $a_k \neq a_{k'}$ for all $k \neq k'$, $b_k \neq b_{k'}$ for all $k \neq k'$, and $a_k \neq b_{k'}$ for all k, k'. For each position not defined by the above rule, we put a unique character which appears only once at the position. Finally, we let $L = m$.

Here we briefly show that this reduction is an L-reduction. Let $\mathcal{I}$ be an instance of Max Cut. Let $\mathcal{I}'$ be the instance produced by the above reduction from $\mathcal{I}$. The score of $\mathcal{I}'$ is given by

$$2 \cdot |\{(t_i[k], t_j[k]) |\ i < j, t_i[k] = t_j[k]\}|$$

because each character can appear at most twice, each character appearing at most once in the same column does not contribute to the score (since $1 \log 1 = 0$), and each character appearing twice in the same column contributes to the score by $2 \log 2 = 2$.

Then, we can see that, given a cut (V_1, V_2) with the score (i.e., the number of edges between V_1 and V_2) x, we can obtain a solution (i.e., an alignment) of $\mathcal{I}'$ with the score $2x$ by letting

$$t_i = s_i[1] \ldots s_i[m] \quad \text{if } v_i \in V_1,$$
$$t_i = s_i[2m+1] \ldots s_i[3m] \quad \text{otherwise.}$$

Moreover, the maximum score of $\mathcal{I}'$ is attained by the solution obtained from the max cut in this way. Therefore, $OPT(\mathcal{I}') = 2OPT(\mathcal{I})$ holds.

Given a solution of $\mathcal{I}'$, we can obtain a cut by the following rule: if $s_i[k]$ appears in t_i for some k such that $1 \leq k \leq m$, then put v_i in V_1, otherwise put v_i in V_2. Then, the score of the obtained cut is at least half of the score of the solution of $\mathcal{I}'$.

Since all the construction can be done in polynomial time, the reduction is an L-reduction and thus the theorem follows. □

Although the LMW algorithm is conceptually simple, we can develop a much simpler PTAS. First note that the maximum #LOG#-score is at most $Ln \log n$, where this case is attained when $t_1 = t_2 = \ldots = t_n$. On the other hand, the minimum #LOG#-score is at least

$$Ln \log(n/A) = Ln(\log n - \log A),$$

where this case is attained when $f_j(a) = \frac{1}{A}$ for all j and for all $a \in \Sigma$. Here, we can see that

$$\frac{Ln(\log n - \log A)}{Ln \log n} > 1 - \epsilon$$

holds if $n > A^{(1/\epsilon)}$. This leads to the following PTAS:

> If $n < A^{(1/\epsilon)}$, then find an optimal local alignment by exhaustive search.
> Otherwise, select an arbitrary substring of length L from each s_i.

Since if $n \geq A^{(1/\epsilon)}$ an arbitrary substring is chosen from each string, solutions obtained by this algorithm may be far from one which captures any feature of sequences. This suggests that #LOG#-score is not adequate to evaluate the local alignment.

4. Results on Relative Entropy Score

Li, Ma and Wang showed in Ref. 15) an upper bound of the difference between the optimal score and the score of the approximate solution that can be found by the LMW algorithm under the relative entropy scoring. However, there is no known result for the worst case *ratio* of a score of an approximate solution to that of the optimal solution. Here, we prove that Local

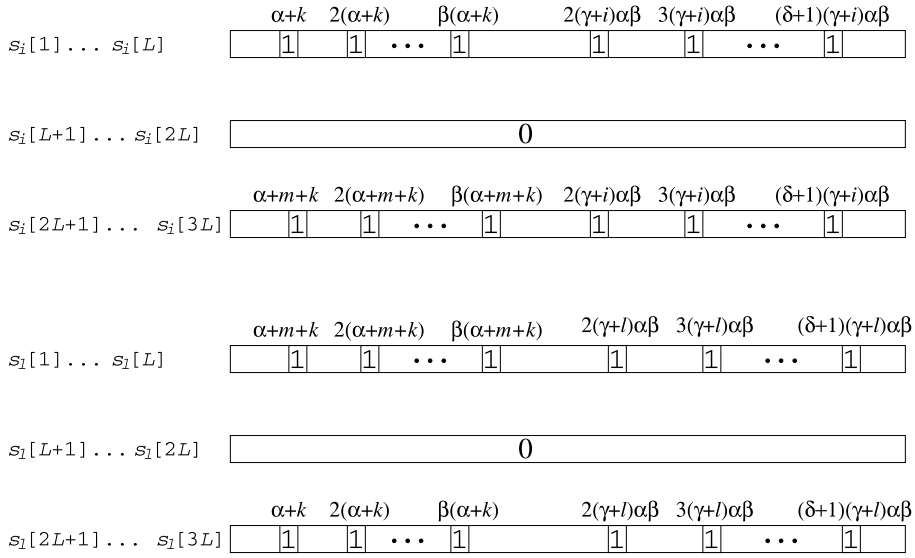

Fig. 2 Example of s_i and s_l such that $e_k = (i, l)$ and $i < l$.

Multiple Alignment under relative entropy scoring is APX-hard even for the binary alphabet.

Theorem 4.1 Local Multiple Alignment under relative entropy scoring is APX-hard even for the binary alphabet.

Proof. We use a PTAS-reduction [3),4)] from Max Cut-B. Max Cut-B is a restriction of Max Cut in which the maximum degree of vertices of the input graph is bounded by B. Max Cut-B is known to be APX-hard even for $B = 3$ [3),16)]. We use a reduction similar to that in Theorem 3.1. But, in this case, a more elaborated reduction is required.

Let $G(V, E)$ be an input graph of Max Cut-B where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We assume w.l.o.g. (without loss of generality) that $m > n$. Let $\Sigma = \{0, 1\}$ and let $p_0 = p_1 = 0.5$.

From $G(V, E)$, we construct $2n + 2$ strings $s_1, \ldots, s_{2n+2}$. Each of $s_1, \ldots, s_n$ has length $3L$ and each of $s_{n+1}, \ldots, s_{2n+2}$ has length L, where L is to be determined later. As in the proof of Theorem 3.1, s_i $(1 \le i \le n)$ corresponds to v_i. $s_{n+1}, \ldots, s_{2n+2}$ are constructed by:

$s_{n+1}[j(\alpha + k)] = s_{n+1}[j(\alpha + m + k)] = 0$
for $j = 1, \ldots, \beta$ and
for $k = 1, \ldots, 2m$,
$s_{n+1}[h] = 1$
otherwise,
$s_{n+2}[j(\gamma + k)\alpha\beta] = 1$
for $j = 2, \ldots, \delta + 1$ and
for $k = 1, \ldots, n$,
$s_{n+2}[h] = 0$
otherwise,
$s_i[h] = 1$
for all h and for $i = n+3, \ldots, 2n+2$,

where α, β, γ, δ are integers to be determined later. Since each of $s_{n+1}, \ldots, s_{2n+2}$ has length L, $s_i = t_i$ should hold for $i = n + 1, \ldots, 2n + 2$. We construct $s_1, \ldots, s_n$ by the following rules (see also **Fig. 2**):

$s_i[j(\alpha + k)] = s_i[2L + j(\alpha + m + k)] =$
$s_l[j(\alpha + m + k)] = s_l[2L + j(\alpha + k)] = 1$
for $j = 1, \ldots, \beta$ if $e_k = \{v_i, v_l\} \in E$ and $i < l$,
$s_i[j(\gamma + i)\alpha\beta] = s_i[2L + j(\gamma + i)\alpha\beta] = 1$
for $j = 2, \ldots, \delta + 1$,
$s_i[h] = 0$
otherwise,

where $L = 2 \cdot \alpha\beta\gamma\delta$. For each s_i $(i = 1, \ldots, n)$, $s_i[1] \ldots s_i[2\alpha\beta\gamma]$ and $s_i[2L+1] \ldots s_i[2L+2\alpha\beta\gamma]$ are called region R_1, and the other parts are called region R_2. In the above construction, $s_i[j(\alpha + k)] = 1$, $s_i[2L + j(\alpha + m + k)] = 1$, $s_l[j(\alpha + m + k)] = 1$ and $s_l[2L + j(\alpha + k)] = 1$ correspond to $s_i[k] = a_k$, $s_i[2m + k] = b_k$, $s_l[k] = b_k$ and $s_l[2m+k] = a_k$ in Theorem 3.1 ☆, respectively. $s_i[j(\gamma+i)\alpha\beta] = 1$ and $s_i[2L+j(\gamma+i)\alpha\beta] = 1$ are used so that either $s_i[1] \ldots s_L[i]$ or $s_i[2L+1] \ldots s_i[3L]$ corresponds to a motif region

☆ In Theorem 3.1, j is used in place of l.

for each $i = 1, \ldots, n$ in the optimal alignment. Let $\mathcal{I}$ denote an instance of MAX CUT-B and let $\mathcal{I}'$ denote the instance of LOCAL MULTIPLE ALIGNMENT constructed from $\mathcal{I}$ as above.

Here, we let $\alpha = 10m\beta$, $\beta = 10n^4$, $\gamma = 10\delta n$, $\delta = 100Bn^4$, where much smaller values might suffice. Then,

$$(j' - j)(\alpha + k) \neq (j''' - j'')(\alpha + k')$$

holds for all j, j', j'', j''' such that $j \neq j'$ and $j'' \neq j'''$ if $k \neq k'$ $(0 < k, k' \leq 2m)$. This can be seen as follows. Let $j_2 = j''' - j''$ and $j_1 = j' - j$. If $j_1 = j_2$, we have

$$j_2(\alpha + k') - j_1(\alpha + k) = j_1(k' - k) \neq 0.$$

Otherwise, we assume w.l.o.g. that $j_2 > j_1$ holds. Then, we have

$$\begin{aligned} &j_2(\alpha + k') - j_1(\alpha + k) \\ &\quad = (j_2 - j_1)\alpha + j_2k' - j_1k \\ &\quad \geq \alpha + j_2k' - j_1k \\ &\quad \geq \alpha - 2\beta m \\ &\quad > 0. \end{aligned}$$

This property guarantees that if $s_i[h] = 1$ is aligned with $s_l[h''] = 1$, then $s_i[h'] = 1$ cannot be aligned with $s_l[h''']$ for any h''', where $1 \leq i \neq l \leq n$ holds and either $1 \leq h, h', h'', h''' \leq 2\alpha\beta\gamma$ or $2L + 1 \leq h, h', h'', h''' \leq 2L + 2\alpha\beta\gamma$ holds.

Similarly,

$$(j' - j)(\gamma + i)\alpha\beta \neq (j''' - j'')(\gamma + i')\alpha\beta$$

holds for all $j, j', j'', j''' > 1$ such that $j' \neq j$ and $j''' \neq j''$ if $i \neq i'$. Furthermore,

$$\begin{aligned} &j'(\gamma + i)\alpha\beta - j(\alpha + k) \\ &\quad \neq j'''(\gamma + i')\alpha\beta - j''(\alpha + k') \end{aligned}$$

holds for all $j, j'' > 0$ and $j', j''' > 1$ if $i \neq i'$, and

$$\begin{aligned} &j'(\gamma + i)\alpha\beta - j(\alpha + k) \\ &\quad \neq j'''(\gamma + i')\alpha\beta - j''(\gamma + i')\alpha\beta \end{aligned}$$

holds for all $j > 0$ and $j', j'', j''' > 1$ if $i' \neq i$. From these inequalities, we can see the following.

Observation 1 For any $i \neq l$ such that $1 \leq i, l \leq n$ and $\{v_i, v_l\} \notin E$, $t_i[h] = t_l[h] = 1$ holds for at most one column h.

Observation 2 If neither $t_i = s_1[1] \ldots s_1[L]$ nor $t_i = s_1[2L + 1] \ldots s_1[3L]$ holds, $t_i[h] = s_{n+2}[h] = 1$ holds for at most one column h.

Given a cut (V_1, V_2), we consider the following alignment:

$$\begin{aligned} t_i &= s_i[1] \ldots s_i[L] && \text{if } v_i \in V_1, \\ t_i &= s_i[2L + 1] \ldots s_i[3L] && \text{otherwise.} \end{aligned}$$

Let $\mathcal{C}(V_1, V_2)$ denote the score of the cut (i.e., the number of edges between V_1 and V_2). Then, the score of this alignment is given by (see also **Table 1**)

$$\frac{1}{L} \cdot (2\beta \cdot \mathcal{C}(V_1, V_2) \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2)),$$

where

$$\begin{aligned} \mathcal{E}(x) = &\left(\frac{n + 1 - x}{2(n + 1)}\right) \log \left(\frac{n + 1 - x}{n + 1}\right) \\ &+ \left(\frac{n + 1 + x}{2(n + 1)}\right) \log \left(\frac{n + 1 + x}{n + 1}\right) \end{aligned}$$

for $x \leq n + 1$, otherwise $\mathcal{E}(x) = 1$. Note that

$$\mathcal{E}(x) \approx \left(\frac{1}{\ln 2}\right) \cdot \left(\frac{x^2}{(n + 1)^2}\right)$$

if $x \ll n$, and $\mathcal{E}(x) \leq 1$ for all x.

Next we assume w.l.o.g. that either $t_i = s_i[1] \ldots s_i[L]$ or $t_i = s_i[2L + 1] \ldots s_i[3L]$ holds for $i = 1, \ldots, n - x$, but does not hold for $i = n - x + 1, \ldots, n$. From these t_i's, we make a partition of V into V_1', V_2', V_3' as follows: put v_i in V_1' if $t_i = s_i[1] \ldots s_i[L]$, put v_i in V_2' if $t_i = s_i[2L + 1] \ldots s_i[3L]$, put v_i in V_3' otherwise. We say that t_i is type j if $v_i \in V_j'$. Then $score(t_1, \ldots, t_n, s_{n+1}, \ldots, s_{2n+2})$ is at most

$$\begin{aligned} \frac{1}{L} \cdot \Big(&x^2 \cdot \mathcal{E}(x + 2) + x \cdot 2B \cdot \beta \cdot \mathcal{E}(1) \\ &+ x \cdot 2\delta \cdot \mathcal{E}(1) + x(n - x) \cdot \mathcal{E}(3) \\ &+ 2\beta \cdot \mathcal{C}(V_1', V_2') \cdot \mathcal{E}(1) \\ &+ (n - x) \cdot \delta \cdot \mathcal{E}(2) \Big). \end{aligned}$$

Each term in the above comes from the following reason (see Table 1 for the types of columns).

$[2\beta \cdot \mathcal{C}(V_1', V_2') \cdot \mathcal{E}(1) + (n - x) \cdot \delta \cdot \mathcal{E}(2)]$
: score corresponding to the cut between V_1' and V_2',

$[x^2 \cdot \mathcal{E}(x + 2)]$
: multiple '1's from t_i's of type 3 appear in each of at most x^2 columns (more precisely, at most $x(x - 1)/2$ columns),

$[x \cdot B \cdot \beta \cdot \mathcal{E}(1)]$
: '1' from R_1 of each t_i of type 3 appears in at most $B \cdot \beta$ columns of types III, IV and VI,

$[x \cdot B \cdot \beta \cdot \mathcal{E}(1)]$
: '1' from R_1 of each t_i of type 3 is missing in at most $B \cdot \beta$ columns of types III and IV,

$[x \cdot \delta \cdot \mathcal{E}(1)]$
: '1' from R_2 of each t_i of type 3 appears in at most δ columns of type VI,

Table 1 Score for each column h in the alignment constructed from a cut $\mathcal{C}$.

	$e_k \in \mathcal{C}$		$e_k \notin \mathcal{C}$			
type	I	II	III	IV	V	VI
h	$j(\alpha+k)$	$j(m+\alpha+k)$	$j(\alpha+k)$	$j(m+\alpha+k)$	$j(\gamma+i)\alpha\beta$	others
$t_1[h]$	0	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_i[h]$	1	0	1	0	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_l[h]$	1	0	0	1	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_n[h]$	0	0	0	0	0	0
$s_{n+1}[h]$	0	0	0	0	1	1
$s_{n+2}[h]$	0	0	0	0	1	0
$s_{n+3}[h]$	1	1	1	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$s_{2n+2}[h]$	1	1	1	1	1	1
score	$(1/L)\mathcal{E}(1)$	$(1/L)\mathcal{E}(1)$	0	0	$(1/L)\mathcal{E}(2)$	0

$[x \cdot \delta \cdot \mathcal{E}(1)]$
'1' from R_2 of each t_i of type 3 is missing in at most δ columns of type V,
$[x(n-x) \cdot \mathcal{E}(3)]$
'1' from each t_i of type 3 appears in at most $n-x$ columns of types I and V.

Here, we replace each of t_i's for $i = n-x+1, \ldots, n$ with $t'_i = s_i[1] \ldots s_i[L]$. Then $score(t_1, \ldots, t_{n-x}, t'_{n-x+1}, \ldots, t'_n, s_{n+1}, \ldots, s_{2n+2})$ is at least

$$\frac{1}{L} \cdot (2\beta \cdot \mathcal{C}(V'_1, V'_2) \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2)).$$

Since

$$\delta \cdot \mathcal{E}(2) > x \cdot \mathcal{E}(x+2) + 2B \cdot \beta \cdot \mathcal{E}(1) + 2\delta \cdot \mathcal{E}(1) + (n-x) \cdot \mathcal{E}(3)$$

holds (recall that $\mathcal{E}(2) \approx 4 \cdot \mathcal{E}(1)$, $\delta = 10B\beta$, $\delta \gg x \cdot (n+1)^2$), we have

$$score(t_1, \ldots, t_n, s_{n+1}, \ldots, s_{2n+2}) \leq score(t_1, \ldots, t_{n-x}, t'_{n-x+1}, \ldots, t'_n, s_{n+1}, \ldots, s_{2n+2}),$$

where we assume that n is sufficiently large. From this, we have the following:

- $OPT(\mathcal{I}') = \frac{1}{L} \cdot (2\beta \cdot OPT(\mathcal{I}) \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2))$,
- Given an alignment $t_1, \ldots, t_{2n+2}$ with the score at least $\frac{1}{L} \cdot (2\beta \cdot y \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2))$, we can construct a cut $(V'_1 \cup V'_3, V'_2)$ with score at least y in polynomial time.

Since $OPT(\mathcal{I}) \geq \frac{m}{2} \geq \frac{n}{2}$, $\delta = 10B \cdot \beta$, and $\mathcal{E}(2) \approx 4 \cdot \mathcal{E}(1)$ hold, we have

$$n \cdot \delta \cdot \mathcal{E}(2) < 81 \cdot B \cdot \beta \cdot OPT(\mathcal{I}) \cdot \mathcal{E}(1).$$

Suppose that there exists an approximation algorithm for Local Multiple Alignment which always outputs a solution with score $y' > (1-\epsilon)OPT(\mathcal{I}')$ for some $\epsilon > 0$. Let

$$y = (L \cdot y' - n \cdot \delta \cdot \mathcal{E}(2))/(2\beta \cdot \mathcal{E}(1)).$$

Then, $y' \geq (1-\epsilon)OPT(\mathcal{I}')$ means

$$2\beta \cdot y \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2) > (1-\epsilon)\,(2\beta \cdot OPT(\mathcal{I}) \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2))$$

and thus we have

$$\begin{aligned} y &> \frac{(1-\epsilon)(2\beta \cdot OPT(\mathcal{I}) \cdot \mathcal{E}(1) + n \cdot \delta \cdot \mathcal{E}(2)) - n \cdot \delta \cdot \mathcal{E}(2)}{2\beta \cdot \mathcal{E}(1)} \\ &> \frac{(1-\epsilon)\,(2\beta \cdot OPT(\mathcal{I}) \cdot \mathcal{E}(1)) - \epsilon \cdot n \cdot \delta \cdot \mathcal{E}(2)}{2\beta \cdot \mathcal{E}(1)} \\ &> \frac{(1-\epsilon(2+81B))) \cdot \beta \cdot OPT(\mathcal{I}) \cdot \mathcal{E}(1)}{2\beta \cdot \mathcal{E}(1)} \\ &= (1-\epsilon(1+(81B/2))) \cdot OPT(\mathcal{I}). \end{aligned}$$

Therefore, the reduction presented above is a PTAS-reduction and thus the theorem follows. □

Corollary 4.2 Local Multiple Alignment over the binary alphabet under #LOG#-scoring is NP-hard.

Although we proved a hardness result, we do not yet succeed to develop an approximation algorithm with guaranteed approximation ratio. We comment here that the LMW algorithm outputs a good approximate alignment when the input sequences have a *strong consensus pattern*. We say that an instance $\mathcal{I}$ of Local Multiple Alignment has a strong consensus pattern if $OPT(\mathcal{I}) > c$ holds, where c is a constant not depending on instance (c may be given by users and may depend on Σ and $p(a)$'s). For example, if $f_i(1) > 0.6$ for at least $0.1L$ positions where $\Sigma = \{0, 1\}$ and $p(0) = p(1) = 0.5$, then the score is always greater than a con-

stant $0.1 \cdot (0.6 \log \frac{0.6}{0.5} + 0.4 \log \frac{0.4}{0.5}) \approx 0.02866$. It seems that it suffices to find strong consensus patterns in most practical cases. Li, Ma and Wang proved that the LMW algorithm always outputs an alignment whose score is less than the optimal score by at most $O((\frac{\log r}{r})^{\frac{1}{3}})$, where r is any fixed (sufficiently large) positive integer. Therefore, the LMW algorithm is a PTAS for instances with strong consensus patterns.

5. Results on SP-score

Many theoretical and practical studies have been done based on SP-score [5),8),20)] though there exists some criticism on SP-score [6)]. Therefore, we consider Local Multiple Alignment under SP-scoring.

Gusfield developed an approximation algorithm for global multiple alignment under SP-scoring [8)]. Slightly modifying his algorithm, we obtain the following approximation algorithm for Local Multiple Alignment under SP-scoring.

(1) For all substrings t_i of s_i and for all substrings t_j of s_j where $|t_i| = |t_j| = L$, compute $d(t_i, t_j) = \sum_{k=1}^{L} dist(t_i[k], t_j[k])$.
(2) For all i and for all substrings t_i of s_i, find $t'_1, \ldots, t'_{i-1}, t'_{i+1}, \ldots, t'_n$ minimizing $\sum_{j \neq i} d(t_i, t'_j)$, where t'_j is a substring of s_j.
(3) Output $(t'_1, \ldots, t'_{i-1}, t_i, t'_{i+1}, \ldots, t'_n)$ minimizing the above value.

We call it the 1-STAR algorithm as in Ref. 5). The following proposition can be proved in the same way as in Ref. 8).

Proposition 5.1 The SP-score of a local alignment obtained by the 1-STAR algorithm is at most the twice of the minimum.

On the other hand, we can prove an NP-hardness result as follows.

Theorem 5.2 Local Multiple Alignment under SP-scoring is NP-hard.

Proof. We reduce MIN-2SAT to Local Multiple Alignment under SP-scoring with $\Sigma = \{0, 1, a\}$. Recall that MIN-2SAT is, given a set of clauses $C = \{c_1, \ldots, c_m\}$ over a set of variables $X = \{x_1, \ldots, x_n\}$ where each c_i consists of at most two literals, to find a truth assignment to X which satisfies the minimum number of clauses [7)].

From an instance of MIN-2SAT, we construct $2n-3$ sequences s_i having the following form

$$s_i = A \cdot B_i \cdot A \cdot D_i \cdot A,$$

where $x \cdot y$ denotes the concatenation of x and y, $|A| = |B_i| = |D_i| = m$, and $A[i] = a$ for all $i = 1, \ldots, m$. For $i = 1, \ldots, n$, B_i is defined by

$$B_i[j] = \begin{cases} 1, & \text{positive literal } x_i \text{ appears in } c_j, \\ 0, & \text{otherwise.} \end{cases}$$

For $i = n+1, \ldots, 2n-3$, B_i is defined by

$$B_i[j] = 1, \quad j = 1, \ldots, m.$$

Similarly, for $i = 1, \ldots, n$, D_i is defined by

$$D_i[j] = \begin{cases} 1, & \text{negative literal } \overline{x}_i \text{ appears in } c_j, \\ 0, & \text{otherwise.} \end{cases}$$

For $i = n+1, \ldots, 2n-3$, D_i is defined by

$$D_i[j] = 1, \quad j = 1, \ldots, m.$$

Here, we let $L = 3m$ and define the distance function by $dist(x, x) = 0$ for $x = 0, 1, a$, $dist(a, x) = dist(x, a) = n^2 m$ for $x = 0, 1$, and $dist(x, y) = 1$ for the other $x \neq y$. By considering the correspondence:

$$x_i = 1 \Longleftrightarrow A \cdot B_i \cdot A \text{ is selected as } t_i,$$
$$x_i = 0 \Longleftrightarrow A \cdot D_i \cdot A \text{ is selected as } t_i,$$

we can see the following property:

- There exists a local multiple alignment with score at most $K(n-2)(n-1) + (m-K)(n-3)n$ if and only if there exists a truth assignment which satisfies at most K clauses.

Since the reduction can be done in polynomial time, we have the theorem. □

6. Concluding Remarks

In this paper, we studied theoretical aspects of Local Multiple Alignment. We proved that Local Multiple Alignment under relative entropy scoring is APX-hard, whereas there exists a PTAS under #LOG#-scoring [15)]. Although these scoring schemes are closely related, there is a large gap on the approximability. The result suggests that the scoring schemes greatly influence the approximability and thus, should be considered as an important factor in approximation algorithms.

Although we proved that Local Multiple Alignment under relative entropy scoring is APX-hard, we do not yet succeed to develop an algorithm with a constant factor approximation ratio. Therefore, development of such an algorithm is an open problem. We employed a *pure* relative entropy score in this paper. However, *pseudocounts* are usually introduced in practice [6),14)]. Therefore, the effect of pseudocounts on the approximability should also be studied.

In practice, the search for non-gapped motifs in biological sequences usually involves motifs of length 5∼25 or so. If longer motifs are needed, gaps should be introduced. On the other hand, the length of motifs used in the proof of Theorem 4.1 is quite large (it is $\Omega(n^{18})$ where n is the number of input sequences). Thus, the result is not important from a practical viewpoint. If the length of a motif is short, we might be able to develop a polynomial time algorithm or a polynomial time approximation scheme. Indeed, it is known that LOCAL MULTIPLE ALIGNMENT can be solved in linear time if the motif length is bound by a constant [10),11)]. Furthermore, Horton and Fujibuchi derived a non-trivial upper bound on the factor depending on motif length and alphabet size in the time complexity [11)]. They posed an interesting open problem which asks the time complexity of LOCAL MULTIPLE ALIGNMENT under a general scoring scheme (including relative entropy scoring) when motif length is $O(\log n)$.

We have also studied LOCAL MULTIPLE ALIGNMENT under SP-scoring. Though it is not useful or important in practice, it is interesting from a theoretical viewpoint since there exists a simple approximation algorithm as shown in this paper. For this problem, there remains a gap between the positive result (approximation factor 2) and the negative result (NP-hardness). Thus, it is also left as an open problem to shorten the gap.

Acknowledgments We would like to thank Paul Horton for valuable discussions. We also thank to Kim Lan Sim for her help.

References

1) Akutsu, T., Arimura, H. and Shimozono, S.: On approximation algorithms for local multiple alignment, *Proc. 4th Int. Conf. Computational Molecular Biology* (*RECOMB 2000*), pp.1–7 (2000).
2) Arora, S., Lund, C., Motwani, R., Sudan, M. and Szegedy, M.: Proof verification and the hardness of approximation problems, *J. ACM*, Vol.45, No.3, pp.501–555 (1998).
3) Ausiello, G., Crescenzi, P. and Protasi, M.: Approximate solution of NP optimization problems, *Theoretical Computer Science*, Vol.150, No.1, pp.1–55 (1995).
4) Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Spaccamela, A.M. and Protasi, M.: *Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties*, Springer-Verlag, Berlin (1999).
5) Bafna, V., Lawler, E.L. and Pevzner, P.: Approximation algorithms for multiple sequence alignment, *Theoretical Computer Science*, Vol.182, No.1/2, pp.233–244 (1997).
6) Durbin, R., Eddy, S., Krough, A. and Mitchison, G.: *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids*, Cambridge Univ. Press, Cambridge, UK (1998).
7) Garey, M.R. and Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., New York (1979).
8) Gusfield, D.: Efficient method for multiple sequence alignment with guaranteed error bounds, *Bulletin of Mathematical Biology*, Vol.55, No.1, pp.141–154 (1993).
9) Horton, P.: A branch and bound algorithm for local multiple alignment, *Proc. Pacific Symp. Biocomputing '96* (*PSB'96*), pp.368–383 (1996).
10) Horton, P.: Tsukuba BB: A branch and bound algorithm for local multiple alignment of DNA and protein sequences, *J. Computational Biology*, Vol.8, No.3, pp.283–303 (2001).
11) Horton, P. and Fujibuchi, W.: An upper bound on the hardness of exact matrix based motif discovery, *Proc. 16th Annual Symposium on Combinatorial Pattern Matching* (*CPM 2005*), *Lecture Notes in Computer Science*, No.3537, pp.219–228 (2005).
12) Hudak, J. and Mcclure, M.A.: A comparative analysis of computational motif-detection methods, *Proc. Pacific Symp. Biocomputing '99* (*PSB'99*), pp.138–149 (1999).
13) Lawrence, C.E. and Reilly, A.A.: An expectation maximization (EM) algorithm for identification and characterization of common sites in unaligned biopolymer sequences, *PROTEINS: Structure, Function, and Genetics*, Vol.7, No.1, pp.41–51 (1990).
14) Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F. and Wootton, J.C.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, *Science*, Vol.262, No.5131, pp.208–214 (1993).
15) Li, M., Ma, B. and Wang, L.: Finding similar regions in many sequences, *J. Computer and System Sciences*, Vol.65, No.1, pp.73–96 (2002).
16) Papadimitriou, C.H. and Yannakakis, M.: Optimization, approximation, and complexity classes, *J. Computer and System Sciences*, Vol.43, No.3, pp.425–440 (1991).
17) Stormo, G. and Hartzell, G.W.: Identifying protein-binding sites from unaligned DNA frag-

ments, *Nucleic Acids Research*, Vol.86, No.4, pp.1183–1187 (1989).
18) Stormo, G.: Consensus patterns in DNA, *Methods in Enzymology*, Vol.183, pp.211–221 (1990).
19) Thompson, J.D., Higgins, D.G. and Gibson, T.J.: CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Research*, Vol.22, No.22, pp.4673–4680 (1994).
20) Wang, L. and Jiang, T.: On the complexity of multiple sequence alignment, *J. Computational Biology*, Vol.1, No.4, pp.337–348 (1994).

(Received December 4, 2006)
(Accepted January 19, 2007)

(Communicated by *Tetsuo Shibuya*)

Tatsuya Akutsu received his M.Eng. degree in Aeronautics in 1996 and a Dr. Eng. degree in Information Engineering in 1989 both from the 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.

Hiroki Arimura received the B.S. degree in 1988 in Physics, the M.S. and the Dr.Sci. degrees in 1990 and 1994 in Information Systems from Kyushu University. From 1990 to 1996, he was at the Department of Artificial Intelligence in Kyushu Institute of Technology, and from 1996 to 2004, he was at the Department of Informatics in Kyushu University. Since 2006, he has been a professor of the Division of Computer Science Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. He has also been an adjunctive researcher with PREST program "Sakigake" of JST for 1999 to 2002. His research interests include data mining, computational learning theory, information retrieval, artificial intelligence, and the design and analysis of algorithms in these fields. He is a member of JSAI, IPSJ, and ACM.

Shinichi Shimozono received Ph.D. in Science from Graduate School of Information Science in Kyushu University in 1996. From 1992, he was a research associate at Kyushu Institute of Technology, and since 1996, he is working as an associate professor of Theoretical Computer Science branch at Department of Artificial Intelligence, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interest includes design and analysis of algorithms for intractable and brand-new computation problems, especially for combinatorial optimization problems that are NP-hard, and analysis of computational hardness of combinatorial problems.