Available online at www.sciencedirect.com
ScienceDirect
Journal of Computer and System Sciences 73 (2007) 997–1011
JOURNAL OF COMPUTER AND SYSTEM SCIENCES
www.elsevier.com/locate/jcss

Near optimal multiple alignment within a band in polynomial time ☆

Bin Ma [a,1], Lusheng Wang [b], Ming Li [c,*,2]

[a] *Department of Computer Science, University of Western Ontario, London, Ontario N6A 5B7, Canada*
[b] *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*
[c] *David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

Received 7 March 2003; received in revised form 10 March 2007

Available online 15 March 2007

Abstract

Multiple sequence alignment is a fundamental problem in computational biology. Because of its notorious difficulties, aligning sequences within a constant band (c-diagonal) is a popular practice in bioinformatics with good practical results [D. Sankoff, J. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Addison–Wesley, 1983; W.R. Pearson, D. Lipman, Improved tools for biological sequence comparison, Proc. Natl. Acad. Sci. USA 85 (1988) 2444–2448; W.R. Pearson, Rapid and sensitive sequence comparison with FASTP and FASTA, Methods Enzymol. 183 (1990) 63–98; W.R. Pearson, Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith–Waterman and FASTA algorithms, Genomics 11 (1991) 635–650; S. Altschul, D. Lipman, Trees, stars, and multiple sequence alignment, SIAM J. Appl. Math. 49 (1989) 197–209; K. Chao, W.R. Pearson, W. Miller, Aligning two sequences within a specified diagonal band, CABIOS 8 (1992) 481–487; J.W. Fickett, Fast optimal alignment, Nucleic Acids Res. 12 (1984) 175–180; E. Ukkonen, Algorithms for approximate string matching, Inform. Control 64 (1985) 100–118; J.L. Spouge, Fast optimal alignment, CABIOS 7 (1991) 1–7]. However, the problem is still NP-hard for multiple sequences. In this paper, we present a theoretical study of this problem. In particular, for arbitrarily small $\epsilon > 0$, we present polynomial time algorithms that produce a multiple alignment (not necessarily c-diagonal) with cost at most $1 + \epsilon$ times the cost of the optimal c-diagonal alignment, under standard models of both SP alignment and consensus (star) alignment. Our algorithms for consensus alignment allow very general score schemes.

In order to prove our main results, we also present similar results for SP alignment and consensus alignment, allowing only constant number of insertion and deletion gaps (of arbitrary length) per sequence on the average. These results are interesting in their own rights and they improve some results in [M. Li, B. Ma, L. Wang, Finding similar regions in many sequences, in: Proc. 31st ACM Symp. Theory of Computing, Atlanta, 1999, pp. 473–482].

Keywords: Multiple sequence alignment within a band; Polynomial time approximation algorithms

☆ A preliminary version has appeared in: Proc. 32nd ACM Symp. Theory Comput., Portland, May 2000.

* Corresponding author.
E-mail addresses: bma@uwo.ca (B. Ma), lwang@cs.cityu.edu.hk (L. Wang), mli@uwaterloo.ca (M. Li).

1 This work was partly done at CityU and Peking University.

2 Part of this work was done while visiting City University of Hong Kong.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2007.03.012

1. Introduction

Multiple sequence alignment is a fundamental and challenging problem in computational molecular biology [1, 4,5,16,23]. It plays a central role in finding similarities and highly conserved subregions among a set of biological sequences. Those conserved subregions may represent some common functions or binding sites.

Different objective functions may be used for measuring the quality of the alignments, in this paper we will only consider the commonly used standard models [5,23]: the *SP alignment* and the *Consensus alignment* (also called Star-alignment). Given sequences $s_1, \ldots, s_n$ over alphabet Σ, let $\mathcal{M}$ be a multiple alignment of these sequences. Any aligned sequence S_i in $\mathcal{M}$ corresponding to s_i is called the *supersequence* of s_i, and can be viewed as a sequence over $\Sigma \cup \{\Delta\}$, where $\Delta \notin \Sigma$ is a new letter indicating a space in S_i. Denote the jth letter of S_i by $S_i[j]$. The majority sequence of $S_1, S_2, \ldots, S_n$ is a sequence S over $\Sigma \cup \{\Delta\}$ such that $S[j]$ is the majority letter of $S_1[j], S_2[j], \ldots, S_n[j]$.

Definition 1 *(SP Alignment).* Find an alignment minimizing the SP-score $\sum_{i \neq j} d_H(S_i, S_j)$, where d_H is the Hamming distance.

Definition 2 *(Consensus Alignment).* Find an alignment minimizing the consensus score $\sum_{i=1}^{n} d_H(S, S_i)$, where S is the majority sequence of $S_1, S_2, \ldots, S_n$ and is called the *median sequence*.

Consensus Alignment is NP-hard for a score scheme where a mismatch costs 1 and a match costs 0 [9]. The problem is MAX SNP-hard if the score scheme is arbitrary [20]. The best known approximation algorithm for Consensus Alignment has performance ratio $2 - o(1)$ [5]. SP alignment has been extensively studied recently. With much effort, the best known performance ratio for SP alignment has been improved from $2 - \frac{2}{k}$ to $2 - \frac{l}{k}$ for any constant l, where k is the number of the sequences [2,4,15]. The $2 - o(1)$ barrier appears to be formidable. There is an enormous literature as well as various other models, methods, and heuristics on multiple sequence alignment for which we refer the reader to [5,23], and [8]. Particularly, Li et al. studied a restricted version of the consensus alignment problem where each given sequence is allowed to have at most c (arbitrarily long) gaps [9]. For constant c, the paper presented a polynomial time algorithm that produces an alignment with cost at most $1 + \epsilon$ times the optimal alignment cost under the c-gap restriction. This result differs from a PTAS (polynomial time approximation scheme) because the output of the algorithm may not satisfy the c-gap requirement of the problem. However, as the c-gap requirement is added mainly for efficiency reason in practice, not satisfying it in the output is not a concern.

In this paper, we are interested in theoretically resolving another popular special case of the multiple alignment problem: *multiple alignment within a band*. The restriction of aligning within c-diagonal band is often applied in many practical cases to reduce the computational complexity. Methods under this assumption have been extensively studied. Sankoff and Kruskal discussed the problem under the rubric of "cutting corners" in [16]. Alignment within a band is used in the final stage of the FASTA program for rapid searching of protein and DNA sequence databases [12,13]. Pearson has shown that alignment within a band gives very good results for lots of protein superfamilies [14]. Other references can be found in [1,3,6,19]. Spouge gives a survey on this topic in [17]. We first define our problem.

Definition 3 *(c-Diagonal Alignment).* Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ be a set of n sequences, each of length m, and $\mathcal{M}$ an alignment of the n sequences. Let the length of the alignment $\mathcal{M}$ be M. A c-diagonal alignment $\mathcal{M}$ is an alignment such that for any p, i and j, if the pth letter of s_i is in column p' of $\mathcal{M}$, and the pth letter of s_j is in column q' of $\mathcal{M}$, then $|p' - q'| \leqslant c$. In other words, the inserted spaces are "evenly" distributed among all sequences and the ith position of a sequence is about at most c positions away from the ith position of any other sequences.

c-Diagonal Alignment remains to be NP-hard in both of the models we consider. The NP-hardness in the consensus alignment model is implied by proofs in [9]. We sketch a proof for the NP-hardness of the SP alignment model in Section 2, Corollary 2. The main results of the paper are the following: for any small $\epsilon > 0$, we have algorithms to compute in polynomial time a multiple sequence alignment with cost at most $1 + \epsilon$ times the cost of an optimal c-diagonal alignment. Different from the conventional definition of PTAS, the output of our algorithms may not satisfy c-diagonal constraint. As discussed before, the c-diagonal restriction is added mainly for efficiency. Therefore, not satisfying it in the output is not a concern. Moreover, as often observed in practice, when an instance is such that the cost of the optimal c-diagonal alignment is very close to the cost of the optimal general alignment, then

```
caaccca
ca cccc
ca cccg
ca ccct
```

Fig. 1. One insertion gap corresponds to three deletion gaps.

our algorithm is a good approximation to the optimal general alignment without the c-diagonal constraint. For these reasons, as well as for the simplicity of presentation, *throughout the paper we use PTAS to denote a polynomial time algorithm that produces a general multiple alignment with at most* $1+\epsilon$ *times the cost of an optimal alignment under the problem definition.* Consequently, the main results of the paper are PTAS for c-DIAGONAL ALIGNMENT under both consensus and SP alignment models.

In order to obtain our main results, we also get similar results for aligning sequences allowing constant number of gaps per sequence on average. These problems are interesting in their own rights. For example, the alignment of Cystic Fibrosis gene (CFTR protein) has only one gap per sequence. Given n sequences $\mathcal{S} = \{s_1, \ldots, s_n\}$, over alphabet $\Sigma = \{1, \ldots, A\}$, to be aligned. Let S_i be the supersequence of s_i in an alignment $\mathcal{M}$, and S be the majority sequence of $S_1, S_2, \ldots, S_n$.

Definition 4 *(INSERTION GAPS AND DELETION GAPS).* If $S[j]$ is Δ while $S_i[j]$ is not, then j corresponds to an insertion of s_i. If $S[j]$ is not Δ while $S_i[j]$ is, then j corresponds to a deletion of s_i. A sequence of consecutive insertions (deletions) is called an insertion gap (a deletion gap).

In *multiple* alignment, one insertion gap may correspond to many deletion gaps and vice versa. Thus, when we count the total number of insertion and deletion gaps, we should do it in an optimal way. For example, in the multiple alignment in Fig. 1, the total number of insertion and deletion gaps should be counted as 1, not 3.

Definition 5 *(AVERAGE c-GAP SP ALIGNMENT).* The AVERAGE c-GAP SP ALIGNMENT problem is to find an alignment of $\mathcal{S}$ such that on average, there are at most c insertion and deletion gaps per sequence, minimizing the SP-score.

Definition 6 *(AVERAGE c-GAP CONSENSUS ALIGNMENT).* The AVERAGE c-GAP CONSENSUS ALIGNMENT problem is to find an alignment of $\mathcal{S}$ such that on the average, there are at most c insertion and deletion gaps per sequence, minimizing the consensus score.

Obviously, we can define the c-GAP SP/CONSENSUS ALIGNMENT problems without the AVERAGE phrase, which require the solution to satisfy that there are at most c insertion and deletion gaps in *every* sequence. Clearly, they are easier versions of the AVERAGE c-GAP SP/CONSENSUS ALIGNMENT problems. In [9], we have constructed a PTAS for c-GAP CONSENSUS ALIGNMENT as a simple application of the polynomial time approximation scheme for CONSENSUS PATTERNS in the same paper. However, it needs brand new approaches in this paper to obtain the PTAS for either the SP model or the c-diagonal model.

The key ideas of our algorithms for the AVERAGE c-GAP CONSENSUS/SP ALIGNMENT problems are as follows: Suppose we randomly pick r letters from n given letters (or from a subset of the n given letters of size $(1-\delta)n$ for a small positive δ), then the frequency of a letter a in the r letters is very close to its frequency in the n letters, with high probability. Moreover, from r random sequences from n sequences (or from a subset of the n sequences of size $(1-\delta)n$ for a small positive δ), we can approximately know the information of the optimal alignment of the n sequences, supposing we know the "correct" alignment of the r sequences. By this approximate information, we can approximately construct the alignment of the n sequences.

The above algorithms are used as subroutines of the algorithms for the c-DIAGONAL CONSENSUS/SP ALIGNMENT problems. Because of the c-diagonal condition, we can dynamically cut the sequences into small segments so that each segments is an Average c-Gap Consensus/SP Alignment and the errors caused by the cutting are small.

2. The hardness results

Theorem 1. 0-GAP SP ALIGNMENT *is NP-hard.*

Proof. W. Just proved that 0-GAP SP ALIGNMENT is NP-hard for the case where the possible gaps are at the two ends of the sequences [7]. The score scheme he used satisfies triangle inequality. However, his result does not imply the NP-hardness for the $w_{a,b} \in \{0, 1\}$ score scheme, where $w_{a,b}$ is the score between the letters a and b.

We reduce MAXIMUM CUT-3 to 0-GAP SP ALIGNMENT. MAXIMUM CUT-3 asks for a maximum cut of a graph G where every node has degree no more than 3. It is known to be Max SNP-hard [11] and hence NP-hard. Let $G = \langle V, E\rangle$ be an instance of MAXIMUM CUT-3, where G is an undirected graph, $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. Let the alphabet $\Sigma = \{0, 1, \mathrm{x}\}$. We first design a sequence s_i for each $v_i \in V$. s_i contains m pieces $p_{i,1}, p_{i,2}, \ldots, p_{i,m}$, each corresponds to an e_j, where $p_{i,j}$ is defined as follows:

$$p_{i,j} = \begin{cases} \mathtt{Xx101xX}, & \text{if } e_j = \langle v_i, v_{i'}\rangle \text{ and } i < i', \\ \mathtt{Xx010xX}, & \text{if } e_j = \langle v_i, v_{i'}\rangle \text{ and } i > i', \\ \mathtt{XxxxxxX}, & \text{otherwise,} \end{cases}$$

$\mathtt{X} = \mathtt{x}^M$ and M is a sufficient large number, e.g., $(nm)^3$. Let $e_j = \langle v_i, v_{i'}\rangle$ and $i < i'$. Let us observe three possible alignments of the jth piece of s_i and $s_{i'}$.

$p_{i,j}$:	Xx101xX	Xx101xX	Xx101xX
$p_{i',j}$:	Xx010xX	Xx010xX	Xx010xX
	(a)	(b)	(c)

Note that $\mathtt{X} = \mathtt{x}^M$ and $p_{k,j} = \mathtt{XxxxxxX}$ for any $k \notin \{i, i'\}$. So, it is easy to verify that if $p_{k,j}$ is aligned with $p_{i,j}$ and $p_{i',j}$, then the columns that contain 0 or 1 have score $3 \times (4n-6) = 12n-18$, $2 \times (2n-2) + 2 \times (4n-8) = 12n-20$ and $2 \times (2n-2) + 2 \times (4n-8) = 12n-20$ for cases (a), (b) and (c), respectively. That is, case (a) has score 2 more than that of case (b) or case (c).

For any $1 \leqslant i \leqslant n$, let S_i be the sequence obtained by concatenating N copies of s_i, where $N = 3n^2$. Let $S_1, S_2, \ldots, S_n$ be the n sequences to be aligned. We get an instance of SP alignment.

For any partition (V_1, V_2) of V with K cut edges, we align the sequences as follows: (1) if $v_i \in V_1$, add a space at the left end of s_i, and (2) if $v_i \in V_2$, add a space at the right end of s_i.

It is easy to verify that the SP-score is $N \times ((m-K) \times (12n-18) + K \times (12n-20)) + 4 \times |V_1| \times |V_2| = Nm(12n-18) - 2NK + 4 \times |V_1| \times |V_2| \leqslant Nm(12n-18) - 2NK + N$. Conversely, given an alignment with SP-score no more than this number, a careful analysis will show that one can get a partition of V that cuts at least K edges. □

In the above proof, since $\mathtt{X} = \mathtt{x}^M$, M is large and $\deg(v) \leqslant 3$ for any $v \in V$, it is not difficult to show that the reduction works for the four versions: SP ALIGNMENT, c-GAP SP ALIGNMENT, AVERAGE c-GAP SP ALIGNMENT and c-DIAGONAL ALIGNMENT of SP score. Hence, we have the following corollary:

Corollary 2. *For any $c \geqslant 0$,* SP ALIGNMENT*, c-*GAP SP ALIGNMENT*,* AVERAGE *c-*GAP SP ALIGNMENT *and c-*DIAGONAL ALIGNMENT *of SP score are all NP-hard.*

3. A PTAS for SP alignment within a band

In this section we prove the following main result: there is a PTAS for c-DIAGONAL ALIGNMENT under SP-score model. The proof consists of two parts. In Section 3.1, we construct a PTAS for the AVERAGE c-GAP SP ALIGNMENT problem; then, using this PTAS, we obtain our main result in Section 3.2.

3.1. AVERAGE *c*-GAP SP ALIGNMENT

Let L be the length of alignment $\mathcal{M}$ of sequences $s_1, \ldots, s_n$. Let $x_{j,a}$ be the number of the occurrences of letter a at the jth position of $\mathcal{M}$. The SP-score of $\mathcal{M}$ can be rewritten as:

$$SP(\mathcal{M}) = \sum_{j=1}^{L} \sum_{\substack{a \neq b \\ a,b \in \Sigma \cup \{\Delta\}}} x_{j,a} x_{j,b}.$$

Algorithm AverageSPAlign

Input: integer c, l, r and $\mathcal{S} = \{s_1, \ldots, s_n\}$, each s_i has length m.
Output: a multiple alignment $\mathcal{M}$.

1. **for** L from m to nm **do**
   for any $s_{i_1}, s_{i_2}, \ldots, s_{i_r} \in \mathcal{S}$ **do**
   for any possible alignment $\mathcal{M}'$ of $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ such that the length is L and each sequence contains no more than cl gaps **do**
   (a) Let $\lambda_{j,a}$ be the number of the occurrences of letter a at the jth position of the alignment $\mathcal{M}'$ for $j = 1, 2, \ldots, L$ and $a \in \Sigma \cup \{\Delta\}$.
   (b) **for** i from 1 to n **do**
   Using dynamic programming to calculate a supersequence S_i of s_i, such that $l(S_i) = L$ and $\sum_{j=1}^{L} \lambda_{j,S_i[j]}$ is maximized.
   (c) Let $\mathcal{M} = \{S_1, S_2, \ldots, S_n\}$ be a multiple alignment of $s_1, s_2, \ldots, s_n$. Calculate $SP(\mathcal{M})$.
2. Output an alignment $\mathcal{M}$ s.t. $SP(\mathcal{M})$ is minimized among all $\mathcal{M}$'s obtained in step 1(c).

Fig. 2. A PTAS for AVERAGE c-GAP SP ALIGNMENT.

Clearly, $\frac{x_{j,a}}{n}$ is the frequency of letter a in the jth position of the alignment. We call the $L \times (|\Sigma| + 1)$ matrix formed by $\frac{x_{j,a}}{n}$ the *frequency matrix* of $\mathcal{M}$.

Our algorithm consists of two major steps: (1) Randomly choose (or trying all possibilities) r sequences from the n sequences. By trying all possible "feasible" alignments, we can suppose that we know the "correct" alignment $\mathcal{M}^r$ of the r sequences that is induced by $\mathcal{M}$. Then we calculate the frequency matrix of $\mathcal{M}^r$, which is hopefully an approximation to the frequency matrix of $\mathcal{M}$. (2) Align every sequence with the frequency matrix of $\mathcal{M}^r$. The complete algorithm is given in Fig. 2.

Theorem 3. *If $l \geqslant 4$, $r \geqslant 1$, Algorithm AverageSPAlign in Fig. 2 outputs an alignment with SP-score no more than $1 + \frac{2}{r} + \frac{2}{l}$ times the SP-score of an optimal* AVERAGE c-GAP SP ALIGNMENT.

Proof. Let $\mathcal{M}_{\text{opt}}$ be an optimal AVERAGE c-GAP SP ALIGNMENT of $s_1, s_2, \ldots, s_n$ and L be the length of $\mathcal{M}_{\text{opt}}$. Let S'_i be the supersequence of s_i in $\mathcal{M}_{\text{opt}}$. For any $l > 0$, since the total number of gaps is no more than cn in $\mathcal{M}_{\text{opt}}$, the number of sequences which contain more than cl gaps is less than $\frac{n}{l}$. Let $\delta = \frac{1}{l}$ and assume, without loss of generality, $\mathcal{S}' = \{s_1, s_2, \ldots, s_{(1-\delta)n}\}$ such that each sequence contains no more than cl gaps.

With equal probability, we randomly choose a sequence from $\mathcal{S}'$ (and put it back). Independently repeat for r times, we get $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$. The alignment $\mathcal{M}_{\text{opt}}$ induces an alignment $\mathcal{M}'$ of $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$, where $\mathcal{M}' = \{S'_{i_1}, S'_{i_2}, \ldots, S'_{i_r}\}$. Starting with these r sequences and the alignment $\mathcal{M}'$, we can get $\lambda_{j,a}$ and $\mathcal{M}$ in steps 1(a)–(c).

Let $\tilde{x}_{j,a} = \lambda_{j,a} \times n/r$. Let $x_{j,a}$ be the number of the occurrences of letter a at the jth position of $\mathcal{M}_{\text{opt}}$, and $y_{j,a}$ be the number of the occurrences of letter a at the jth position of $\mathcal{M}$. Then to prove the theorem, we only need to prove that

$$E\left[\sum_{j=1}^{L} \sum_{\substack{a \neq b \\ a,b \in \Sigma \cup \{\Delta\}}} y_{j,a} y_{j,b}\right] \leqslant \left(1 + \frac{2}{r} + 2\delta\right) \sum_{j=1}^{L} \sum_{\substack{a \neq b \\ a,b \in \Sigma \cup \{\Delta\}}} x_{j,a} x_{j,b}. \tag{1}$$

We prove Inequality (1) via several claims.

Claim 4. *Let $\delta_1, \delta_2, \ldots, \delta_k$ be k numbers with $\sum_{i=1}^{k} \delta_i = 0$. Then $\sum_{\substack{1 \leqslant i,j \leqslant k \\ i \neq j}} \delta_i \delta_j = -\sum_{i=1}^{k} \delta_i^2 \leqslant 0$.*

Proof. Since $(\sum_{i=1}^{k} \delta_i)^2 = 0$, so $\sum_{i=1}^{k} \delta_i^2 + \sum_{i \neq j} \delta_i \delta_j = 0$. The claim follows. □

Claim 5. $\sum_{j=1}^{L} \sum_{a \neq b} y_{j,a} y_{j,b} \leqslant 2 \sum_{j=1}^{L} \sum_{a \neq b} \tilde{x}_{j,a} y_{j,b} - \sum_{j=1}^{L} \sum_{a \neq b} \tilde{x}_{j,a} \tilde{x}_{j,b}$.

Proof. It is sufficient to show that

$$\sum_{j=1}^{L}\sum_{a\neq b}(y_{j,a}-\tilde{x}_{j,a})(y_{j,b}-\tilde{x}_{j,b})\leqslant 0.$$

This follows from Claim 4 and $\sum_{a\in\Sigma\cup\{\Delta\}}(y_{j,a}-\tilde{x}_{j,a})=0$. □

$\mathcal{M}$ is constructed to optimize the alignment to $\mathcal{M}'$, whereas $\mathcal{M}_{\text{opt}}$ is to optimize the alignment of all sequences. Intuitively, the values $y_{j,b}$ from $\mathcal{M}$ is more "compatible" to $\tilde{x}_{j,a}$ than $x_{j,b}$ is. We have the following claim.

Claim 6. $\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}y_{j,b}\leqslant\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}x_{j,b}$.

Proof. Let S_i and S_i' be the supersequences of s_i in $\mathcal{M}$ and $\mathcal{M}_{\text{opt}}$, respectively. Let $\chi(a,b)=0$ if $a\neq b$ and $\chi(a,b)=1$ if $a=b$. Then

$$\begin{aligned}\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}y_{j,b}&=\sum_{j=1}^{L}\sum_{a\in\Sigma\cup\{\Delta\}}\tilde{x}_{j,a}(n-y_{j,a})=n\sum_{j=1}^{L}\sum_{a\in\Sigma\cup\{\Delta\}}\tilde{x}_{j,a}-\sum_{j=1}^{L}\sum_{a\in\Sigma\cup\{\Delta\}}\sum_{i=1}^{n}\chi\big(S_i[j],a\big)\tilde{x}_{j,a}\\&=n^2L-\frac{n}{r}\times\sum_{i=1}^{n}\sum_{j=1}^{L}\sum_{a\in\Sigma\cup\{\Delta\}}\chi\big(S_i[j],a\big)\lambda_{j,a}=n^2L-\frac{n}{r}\times\sum_{i=1}^{n}\sum_{j=1}^{L}\lambda_{j,S_i[j]}.\end{aligned}$$

For the same reason,

$$\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}x_{j,b}=n^2L-\frac{n}{r}\times\sum_{i=1}^{n}\sum_{j=1}^{L}\lambda_{j,S_i'[j]}.$$

From step 1(b), we know that $\sum_{j=1}^{L}\lambda_{j,S_i[j]}\geqslant\sum_{j=1}^{L}\lambda_{j,S_i'[j]}$. Therefore, the claim is proved. □

With Claims 5 and 6, in order to prove (1), it is only necessary to prove the following claim.

Claim 7.

$$E\left[2\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}\times x_{j,b}-\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}\times\tilde{x}_{j,b}\right]\leqslant\left(1+\frac{2}{r}+2\delta\right)\sum_{j=1}^{L}\sum_{a\neq b}x_{j,a}x_{j,b}.$$

Proof. It is sufficient to show that

$$-E\left[\sum_{j=1}^{L}\sum_{a\neq b}(\tilde{x}_{j,a}-x_{j,a})(\tilde{x}_{j,b}-x_{j,b})\right]\leqslant\left(\frac{2}{r}+2\delta\right)\sum_{j=1}^{L}\sum_{a\neq b}x_{j,a}x_{j,b}.\tag{2}$$

Since $\sum_{a\in\Sigma\cup\{\Delta\}}(\tilde{x}_{j,a}-x_{j,a})=0$, by Claim 4, we know that

$$\sum_{a\neq b}(\tilde{x}_{j,a}-x_{j,a})(\tilde{x}_{j,b}-x_{j,b})=-\sum_{a\in\Sigma\cup\{\Delta\}}(\tilde{x}_{j,a}-x_{j,a})^2.$$

Thus, to prove Formula (2), it is sufficient to prove

$$E\left[\sum_{a\in\Sigma\cup\{\Delta\}}(\tilde{x}_{j,a}-x_{j,a})^2\right]\leqslant\left(\frac{2}{r}+2\delta\right)\sum_{a\neq b}x_{j,a}x_{j,b}=\left(\frac{2}{r}+2\delta\right)\sum_{a\in\Sigma\cup\{\Delta\}}x_{j,a}(n-x_{j,a}).\tag{3}$$

Let S_i' be the supersequence of $s_i\in\mathcal{S}'$ for $i=1,2,\ldots,(1-\delta)n$ in $\mathcal{M}_{\text{opt}}$. Let $x'_{j,a}$ be the number of occurrences of letter a in $S_1'[j], S_2'[j],\ldots,S_{(1-\delta)n}'[j]$. Let $x''_{j,a}=\frac{1}{1-\delta}x'_{j,a}$. A moment of thinking shows that $\lambda_{j,a}$ is the sum of r

independent 0–1 random variables, each taking 1 with probability $\frac{x''_{j,a}}{n}$, i.e., $\lambda_{j,a}$ has a binomial distribution $B(r, \frac{x''_{j,a}}{n})$. By a simple property of binomial distribution [18],

$$\mathbf{var}(\lambda_{j,a}) = E\left[\left(\lambda_{j,a} - r \times \frac{x''_{j,a}}{n}\right)^2\right] = r \times \frac{x''_{j,a}}{n} \times \left(1 - \frac{x''_{j,a}}{n}\right). \tag{4}$$

Multiplying Formula (4) by $(\frac{n}{r})^2$, we get

$$E\left[\left(\tilde{x}_{j,a} - x''_{j,a}\right)^2\right] = \frac{1}{r} x''_{j,a}\left(n - x''_{j,a}\right). \tag{5}$$

Obviously, $x''_{j,a} = E[\tilde{x}_{j,a}]$. So, it is easy to verify that

$$E\left[(\tilde{x}_{j,a} - x_{j,a})^2\right] = \left(x_{j,a} - x''_{j,a}\right)^2 + E\left[\left(\tilde{x}_{j,a} - x''_{j,a}\right)^2\right] = \left(x_{j,a} - x''_{j,a}\right)^2 + \frac{1}{r} x''_{j,a}\left(n - x''_{j,a}\right). \tag{6}$$

The last equality comes from Formula (5).

Now let us upper bound the right side of Formula (6). First, let us consider $(x_{j,a} - x''_{j,a})^2$.

Case 1. If $x_{j,a} \leqslant x''_{j,a}$, then we have

$$\left(x_{j,a} - x''_{j,a}\right)^2 \leqslant \left(\frac{x'_{j,a}}{1-\delta} - x_{j,a}\right)\left(x''_{j,a} - x_{j,a}\right) \leqslant \frac{1}{1-\delta}\left(x'_{j,a} - x_{j,a} + \delta x_{j,a}\right)(n - x_{j,a}) \leqslant \frac{\delta}{1-\delta} x_{j,a}(n - x_{j,a}).$$

The last inequality holds since by definition $x'_{j,a} \leqslant x_{j,a}$.

Case 2. If $x_{j,a} > x''_{j,a}$, we still have

$$\begin{aligned}\left(x_{j,a} - x''_{j,a}\right)^2 &\leqslant \left(x_{j,a} - x''_{j,a}\right)\left(x_{j,a} - \frac{x'_{j,a}}{1-\delta}\right) \leqslant \frac{1}{1-\delta}\left(x_{j,a} - x''_{j,a}\right)\left(x_{j,a} - x'_{j,a} - \delta x_{j,a}\right) \\ &\leqslant \frac{1}{1-\delta} x_{j,a}(\delta n - \delta x_{j,a}) = \frac{\delta}{1-\delta} x_{j,a}(n - x_{j,a}).\end{aligned}$$

Therefore, we have

$$\left(x_{j,a} - x''_{j,a}\right)^2 \leqslant \frac{\delta}{1-\delta} x_{j,a}(n - x_{j,a}). \tag{7}$$

Secondly,

$$x''_{j,a}\left(n - x''_{j,a}\right) = \frac{1}{(1-\delta)^2} x'_{j,a}\left((1-\delta)n - x'_{j,a}\right) \leqslant \frac{1}{(1-\delta)^2} x_{j,a}\left((1-\delta)n - x'_{j,a}\right).$$

Since $x_{j,a} - x'_{j,a} \leqslant \delta n$, we have

$$x''_{j,a}\left(n - x''_{j,a}\right) \leqslant \frac{1}{(1-\delta)^2} x_{j,a}(n - x_{j,a}). \tag{8}$$

Combining Formulas (6)–(8), we know that

$$E\left[(\tilde{x}_{j,a} - x_{j,a})^2\right] \leqslant \left(\frac{\delta}{1-\delta} + \frac{1}{r(1-\delta)^2}\right) x_{j,a}(n - x_{j,a}).$$

When $l \geqslant 4$, i.e., $\delta \leqslant \frac{1}{4}$, we have

$$E\left[(\tilde{x}_{j,a} - x_{j,a})^2\right] \leqslant \left(2\delta + \frac{2}{r}\right) x_{j,a}(n - x_{j,a}).$$

Thus, we have proved Formula (3), hence the claim. □

Combining Claims 5–7, Formula (1) is proved, hence the theorem. □

Algorithm DiagonalSPAlign

Input: integers c, l, r and t, and $\mathcal{S} = \{s_1, \ldots, s_n\}$, each s_i has length m.
Output: a multiple alignment $\mathcal{M}$ of $\{s_1, \ldots, s_n\}$.

1. **for** i from 1 to m **do**
   let c_i be the cost of the output of Algorithm AverageSPAlign with input: ct, l, r, and $\{s_1[1..i], s_2[1..i], \ldots, s_n[1..i]\}$.
2. Let L be the maximum i such that $c_i \leqslant \rho ctn^2$. Remove the first L letters from each s_i as a segment.
3. Repeat steps 1, 2 and 3 until every sequence is of length 0.
4. Concatenate the alignments of the segments to form a multiple alignment for the original $\mathcal{S}$.

Fig. 3. A PTAS for c-DIAGONAL ALIGNMENT in SP model.

Remark. If there are weights $w_{a,b} > 0$ satisfying $w_{a,b} = w_a \times w_b$ for $a, b \in \Sigma \cup \{\Delta\}$, our algorithm and theorem still hold for the weighted SP-score defined below:

$$SP(\mathcal{M}) = \sum_{j=1}^{L} \sum_{\substack{a \neq b \\ a,b \in \Sigma}} w_{a,b} x_{j,a} x_{j,b}.$$

We used $w_{a,b} = 1$ for any $a, b \in \Sigma \cup \{\Delta\}$ in our proofs for readability reasons.

3.2. c-DIAGONAL ALIGNMENT *in SP model*

The basic ideas of our c-DIAGONAL ALIGNMENT algorithm are: (1) Dynamically cut the sequences into small segments such that the SP alignment cost for each segment is about ctn^2. Therefore, there are about ct insertions and deletions in a segment of each sequence on average. Thus, we can use the PTAS for AVERAGE c-GAP SP ALIGNMENT for each segment; (2) From the c-diagonal condition, the cost of the cutting errors for every segment is at most cn^2, that is small with respect to ctn^2—the cost of a segment.

Let $\rho = 1 + \frac{2}{l} + \frac{2}{r}$. The complete algorithm is given in Fig. 3.

Theorem 8. *The performance ratio of Algorithm DiagonalSPAlign is* $\rho(1 + \frac{2}{t-2-\frac{1}{c}})$.

Proof. Let $\mathcal{M}_{\text{opt}}$ be an optimal alignment of $s_1, s_2, \ldots, s_n$. Suppose $S_1, S_2, \ldots, S_n$ are the supersequences and c_{opt} is the optimal cost. Let $f_i : [1..m] \to [1..M]$ be the strictly incremental function such that $S_i[f_i(j)] = s_i[j]$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

Suppose in step 4 we get h segments $[j_1, j'_1], [j_2, j'_2], \ldots, [j_h, j'_h]$, where $j_1 = 1$, $j'_h = m$ and $j_i = j'_{i-1} + 1$. For each segment $[j_i, j'_i]$, let $c_{\text{opt}}(i)$ be the optimal alignment cost of $s_1[j_i, j'_i], s_2[j_i, j'_i], \ldots, s_n[j_i, j'_i]$ and $c_{\text{alg}}(i)$ be the alignment cost from Algorithm AverageSPAlign. Then we have

Claim 9. $c_{\text{alg}}(i) \leqslant \rho c_{\text{opt}}(i)$.

Proof. If $c_{\text{opt}}(i) \geqslant ctn^2$, then from the selection of L in step 2, $c_{\text{alg}}(i) \leqslant \rho ctn^2$. So we have $c_{\text{alg}}(i) \leqslant \rho c_{\text{opt}}(i)$. If $c_{\text{opt}}(i) < ctn^2$, it is easy to see that each insertion or deletion contributes score at least n. So, in the optimal alignment for segment i, the average number of insertions and deletions for each sequence is no more than ct. Thus, from Theorem 3, we also have $c_{\text{alg}}(i) \leqslant \rho c_{\text{opt}}(i)$. □

If $h = 1$, the theorem is trivially true by Claim 9. So, in the rest of the proof, we assume $h \geqslant 2$. Let $c(i)$ be the cost in $\mathcal{M}_{\text{opt}}$ contributed by the ith segment. That is,

$$c(i) = \sum_{k=1}^{n} \sum_{j=f_k(j_i)}^{f_k(j'_i)} (n - x_{j,S_k[j]}),$$

where $x_{j,a}$ is the number of the occurrences of letter a at the jth position of $\mathcal{M}_{\text{opt}}$. Then we have

Claim 10. *For any* $1 \leqslant i < h$, $c(i) \geqslant c(t-2)n^2 - n^2$.

Proof. We prove it by contradiction. Suppose that $c(i) < c(t-2)n^2 - n^2$ for some $1 \leqslant i < h$. Then since the c-diagonal condition, the alignment of the ith segment which is induced by $\mathcal{M}_{\text{opt}}$ has score less than $c(i) + 2cn^2$. Therefore, $c_{\text{opt}}(i) < ctn^2 - n^2$. So, the cost of an optimal alignment for $\mathcal{S}' = \{s_1[j_i, j_i' + 1], s_2[j_i, j_i' + 1], \ldots, s_n[j_i, j_i' + 1]\}$ is not more than $c_{\text{opt}}(i) + n^2 < ctn^2$. Thus, by Theorem 3, we know that Algorithm AverageSPAlign will output an alignment for $\mathcal{S}'$ with cost less than ρctn^2. This is a contradiction with the maximal of L in step 2. □

As a consequence of Claim 10, we have

$$c_{\text{opt}} \geqslant \sum_{i=1}^{h-1} c(i) \geqslant (h-1)\big(c(t-2)n^2 - n^2\big). \tag{9}$$

It is easy to see: $c_{\text{opt}}(1) \leqslant c(1) + cn^2$, $c_{\text{opt}}(h) \leqslant c(h) + cn^2$ and $c_{\text{opt}}(i) \leqslant c(i) + 2cn^2$ for $1 < i < h$. So

$$\sum_{i=1}^{h} c_{\text{opt}}(i) \leqslant \sum_{i=1}^{h} c(i) + 2(h-1)cn^2 \leqslant c_{\text{opt}} + 2(h-1)cn^2.$$

Combining with Claim 9, we have

$$\sum_{i=1}^{h} c_{\text{alg}}(i) \leqslant \rho \sum_{i=1}^{h} c_{\text{opt}}(i) \leqslant \rho c_{\text{opt}} + 2\rho(h-1)cn^2. \tag{10}$$

Combining with Formula (9), we have the following which proves the theorem:

$$\frac{\sum_{i=1}^{h} c_{\text{alg}}(i)}{c_{\text{opt}}} \leqslant \rho + \frac{2\rho(h-1)cn^2}{(h-1)(c(t-2)n^2 - n^2)} \leqslant \rho\left(1 + \frac{2}{t - 2 - \frac{1}{c}}\right). \qquad \square$$

4. A PTAS for consensus alignment within a band

In this section we prove the following main result: there is a PTAS for c-DIAGONAL ALIGNMENT in the consensus model. Similar to the SP model, the proof consists of two parts. In Section 4.1, we construct a PTAS for the AVERAGE c-GAP CONSENSUS ALIGNMENT problem. Then, in Section 4.2, using the PTAS in Section 4.1 as a subroutine, we obtain our main result.

4.1. AVERAGE c-GAP CONSENSUS ALIGNMENT

In this section, we design a PTAS for AVERAGE c-GAP CONSENSUS ALIGNMENT. This algorithm itself is a major improvement of a previous weaker result (without average) in [9]. In [9], we proved that the majority letter (sequence) of r random selected letters (sequences) is a good approximation to the majority of n given letters (sequences). Using this property, we presented PTAS to CONSENSUS PATTERNS and c-GAP CONSENSUS ALIGNMENT. Here, we prove a stronger property, that is, if the r random letters are selected from a subset of the n given letters (sequences), and the subset is of size $(1-\delta)n$ for a small positive δ, then the above property still holds (Lemma 13). Using this stronger property, we design the PTAS for average c-gap consensus alignment.

Theorem 11. *For $l > 2$, $r > 2$, Algorithm AverageConsensusAlign gives an alignment with cost at most*

$$1 + \max\left\{\frac{4}{l-2}, \frac{8}{\sqrt{e}(\sqrt{4r+1} - 3)}\right\} A$$

times that of the optimum in polynomial time, where A is the alphabet size.

Proof *(Sketch).* The following technical lemma was proved in [9].

Algorithm AverageConsensusAlign

Input: integers c, l, r, and $\mathcal{S} = \{s_1, \ldots, s_n\}$, each s_i has length m.
Output: a multiple alignment $\mathcal{M}$.

1. **for** L from m to nm **do**
   for any $s_{i_1}, s_{i_2}, \ldots, s_{i_r} \in \mathcal{S}$ **do**
   for any possible alignment $\mathcal{M}'$ of $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ such that the length is L and each sequence contains no more than cl gaps **do**
   (a) Let S be the majority sequence of the supersequences of $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ in $\mathcal{M}'$.
   (b) **for** i from 1 to n **do**
   Using dynamic programming to calculate a supersequence S_i of s_i, such that $l(S_i) = L$ and $d_H(S, S_i)$ is minimized.
   (c) Let $\mathcal{M} = \{S_1, S_2, \ldots, S_n\}$. Calculate the consensus score $\sum_{i=1}^{n} d_H(S, S_i)$.
2. Output an alignment $\mathcal{M}$ s.t. the consensus score is minimized among all $\mathcal{M}$'s obtained in step 1(c).

Fig. 4. PTAS for AVERAGE c-GAP CONSENSUS ALIGNMENT.

Lemma 12. *Let* $g(x, y) = \frac{1}{1-x}(x - y)(1 - x - y + 2\sqrt{xy})^r$. *If* $r \geqslant 3$, $0 \leqslant y < x$ *and* $x + y \leqslant 1$, *then* $g(x, y) < \frac{4}{\sqrt{e}\,(\sqrt{4r+1}-3)}$.

From Lemma 12, we want to prove the following lemma:

Lemma 13. *Let* $a_1, a_2, \ldots, a_n \in \Sigma$ *be n letters. Let* $h(a)$ *be the number of occurrences of letter* a *in* $a_1, a_2, \ldots, a_n$. *For* $1 \leqslant i_1, i_2, \ldots, i_r \leqslant n$, *let* $a_{(i_1,i_2,\ldots,i_r)}$ *be a majority letter of* $a_{i_1}, a_{i_2}, \ldots, a_{i_r}$ *and* a^* *be a majority letter of* $a_1, a_2, \ldots, a_n$. *Let* $k = (1 - 1/l)n$. *For any* $J = \{j_1, j_2, \ldots, j_k\}$, *a subset of* $\{1, 2, \ldots, n\}$, *if* $r \geqslant 3$ *and* $l > 2$, *then*

$$k^{-r} \sum_{i_1,i_2,\ldots,i_r \in J} \big[h(a^*) - h(a_{(i_1,i_2,\ldots,i_r)})\big] \leqslant \max\left\{\frac{4}{l-2}, \frac{8}{\sqrt{e}\,(\sqrt{4r+1}-3)}\right\}(A-1)\big(n - h(a^*)\big). \tag{11}$$

Proof. For every $a \in \Sigma = \{1, \ldots, A\}$, let l_a denote the number of a's in an r-element set. Let

$$J^r = \big\{(i_1, i_2, \ldots, i_r) \mid i_j \in J\big\}$$

be the set of r-tuples of indexes. To simplify the proof, we first introduce two index sets, $\mathcal{I}_a$ and $\mathcal{L}_a$, where

$$\mathcal{I}_a = \big\{(i_1, i_2, \ldots, i_r) \in J^r \;\big|\; a \text{ is a majority of } a_{i_1}, a_{i_2}, \ldots, a_{i_r}\big\},$$
$$\mathcal{L}_a = \big\{(l_1, l_2, \ldots, l_A) \;\big|\; l_1 + l_2 + \cdots + l_A = r \text{ and } l_b \leqslant l_a \text{ for any } b \in \Sigma\big\}.$$

Since $\Sigma = \{1, 2, \ldots, A\}$, then the left-hand side of Inequality (11) is

$$\begin{aligned} k^{-r} \sum_{(i_1,i_2,\ldots,i_r)\in J^r} \big[h(a^*) - h(a_{(i_1,i_2,\ldots,i_r)})\big] &= k^{-r} \sum_{a=1}^{A} \sum_{(i_1,i_2,\ldots,i_r)\in\mathcal{I}_a} \big[h(a^*) - h(a)\big] \\ &= k^{-r} \sum_{a=1}^{A} \big[h(a^*) - h(a)\big]|\mathcal{I}_a|. \end{aligned} \tag{12}$$

To upper bound the left-hand side of Inequality (11), we consider the upper bounds of

$$k^{-r}\big[h(a^*) - h(a)\big]|\mathcal{I}_a|.$$

If $a \neq a^*$, two cases arise:

Case 1. $h(a^*) - h(a) \leqslant 2n/l$.

Obviously, $h(a^*) + h(a) \leqslant n$. Add the two inequalities, we have $2h(a^*) \leqslant (1 + \frac{2}{l})n$. So, $2(n - h(a^*)) \geqslant (1 - \frac{2}{l})n$. Therefore, $\frac{4}{l-2}(n - h(a^*)) \geqslant 2n/l$. Thus, $h(a^*) - h(a) \leqslant 2n/l \leqslant \frac{4}{l-2}(n - h(a^*))$. It is easy to see that $|\mathcal{I}_a|k^{-r} \leqslant 1$. Therefore, we have

$$k^{-r}\big[h(a^*) - h(a)\big]|\mathcal{I}_a| \leqslant \frac{4}{l-2}\big(n - h(a^*)\big). \tag{13}$$

Case 2. $h(a^*) - h(a) > 2n/l$.

Let x_a be the number of occurrences of letter a in $A_J = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$. Then for any $a \in \Sigma$, $x_{a^*} \geqslant h(a^*) - n/l \geqslant h(a^*) - 2n/l \geqslant h(a) \geqslant x_a$. That is, a^* also appears the most in A_J. By a simple counting, we know that the size of set $\mathcal{I}_a$ is

$$|\mathcal{I}_a| = \sum_{(l_1,l_2,\ldots,l_A)\in\mathcal{L}_a} \frac{r!}{l_1!l_2!\cdots l_A!} x_1^{l_1} x_2^{l_2} \cdots x_A^{l_A}. \tag{14}$$

For any $(l_1, l_2, \ldots, l_A) \in \mathcal{L}_a$, since $l_{a^*} \leqslant l_a$, and $x_{a^*} \geqslant x_a$, we have $x_a^{l_a} x_{a^*}^{l_{a^*}} \leqslant (\sqrt{x_a x_{a^*}})^{l_a} (\sqrt{x_a x_{a^*}})^{l_{a^*}}$. Thus, by setting $y_a = y_{a^*} = \sqrt{x_a x_{a^*}}$, and $y_i = x_i$ for $i \neq a$ and $i \neq a^*$, we know that

$$\begin{aligned}
\sum_{(l_1,l_2,\ldots,l_A)\in\mathcal{L}_a} \frac{r!}{l_1!l_2!\cdots l_A!} x_1^{l_1} x_2^{l_2} \cdots x_A^{l_A} &\leqslant \sum_{(l_1,l_2,\ldots,l_A)\in\mathcal{L}_a} \frac{r!}{l_1!l_2!\cdots l_A!} y_1^{l_1} y_2^{l_2} \cdots y_A^{l_A} \\
&\leqslant \sum_{l_1+l_2+\cdots+l_A=r} \frac{r!}{l_1!l_2!\cdots l_A!} y_1^{l_1} y_2^{l_2} \cdots y_A^{l_A} \\
&= (y_1 + y_2 + \cdots + y_A)^r = (k - x_{a^*} - x_a + 2\sqrt{x_{a^*} x_a})^r.
\end{aligned}$$

Combining with Formula (14) and Lemma 12,

$$\begin{aligned}
|\mathcal{I}_a| &\leqslant (k - x_{a^*} - x_a + 2\sqrt{x_{a^*} x_a})^r = k^r \times \left(1 - \frac{x_{a^*}}{k} - \frac{x_a}{k} + 2\sqrt{\frac{x_{a^*}}{k}\frac{x_a}{k}}\right)^r \\
&\leqslant k^r \times \frac{k - x_{a^*}}{x_{a^*} - x_a} \times \frac{4}{\sqrt{e}\,(\sqrt{4r+1} - 3)}.
\end{aligned} \tag{15}$$

It is easy to see that $k - x_{a^*} \leqslant n - h(a^*)$. Since $h(a^*) - h(a) > 2n/l$, we have

$$\frac{k - x_{a^*}}{n - h(a^*)} \times \frac{h(a^*) - h(a)}{x_{a^*} - x_a} \leqslant 1 \times \frac{h(a^*) - h(a)}{h(a^*) - h(a) - n/l} \leqslant 2.$$

Combining with Formula (15), we have

$$|\mathcal{I}_a| \leqslant k^r \times \frac{n - h(a^*)}{h(a^*) - h(a)} \times \frac{8}{\sqrt{e}\,(\sqrt{4r+1} - 3)}.$$

So,

$$k^{-r}\big[h(a^*) - h(a)\big]|\mathcal{I}_a| \leqslant \frac{8}{\sqrt{e}\,(\sqrt{4r+1} - 3)}\big(n - h(a^*)\big). \tag{16}$$

If $a = a^*$, then $h(a^*) - h(a) = 0$. Combining Formulas (12), (13) and (16), we have the lemma. □

Lemma 13 actually shows that $h(a_{i_1,i_2,\ldots,i_r})$ is very close to $h(a^*)$ in expectation. Let S_i be the supersequence of s_i in an optimal alignment. Since the expectation is additive, when we regard the majority sequence of r random sequences from $S_{j_1}, S_{j_2}, \ldots, S_{j_k}$ as the majority sequence of $S_1, S_2, \ldots, S_n$, i.e., the median sequence of the optimal alignment, we get in small error with respect to the optimal alignment cost.

Now let J be the set of j's such that s_j contains at most cl insertions and deletions in an optimal multiple alignment. Since the average number in one sequence is no more than c in an optimal alignment, we know that $|J| \geqslant (1 - 1/l)n$. From Lemma 13, we know that the performance ratio of the Algorithm AverageConsensusAlign is

$$1 + \max\left\{\frac{4}{l-2}, \frac{8}{\sqrt{e}\,(\sqrt{4r+1} - 3)}\right\} A.$$

Note that, in above formula, there is a term A instead of $A - 1$ since the alphabet for S_i's is $\Sigma \cup \{\Delta\}$. □

Algorithm DiagonalAlign

Input: integers c, l, r and t, and $\mathcal{S} = \{s_1, \ldots, s_n\}$, each s_i has length m.
Output: a multiple alignment of $\mathcal{S}$.

1. **for** i from 1 to the length of s_1 **do**
   let c_i be the cost of the output of Algorithm AverageConsensusAlign with input: ct, l, r, and $\{s_1[1..i], s_2[1..i], \ldots, s_n[1..i]\}$.
2. Let L be the maximum i such that $c_i \leqslant \rho ctn$. Remove the first L letters from each s_i as a segment.
3. Repeat steps 1, 2 and 3 until every sequence is of length 0.
4. Put the alignments of the segments together to get an multiple alignment for the original $\mathcal{S}$.

Fig. 5. A PTAS for c-DIAGONAL ALIGNMENT in consensus model.

4.2. c-DIAGONAL ALIGNMENT *in consensus model*

We now present a PTAS for c-DIAGONAL ALIGNMENT under consensus alignment model. The algorithm is almost the same to our PTAS for SP model: (1) Dynamically cut the n sequences into small segments such that the total alignment cost for each segment is about ctn for some constant t, i.e., about ct for each piece. That is, there are about ct insertions and deletions for each piece on average. (2) Since the c-diagonal condition, each cut brings in $O(cn)$ error. Thus, the parameter t acts against the errors taken in by the uncertainty of the cutting. (3) Use Algorithm AverageConsensusAlign for each segment and put the segments together.

Let $\rho = 1 + \max\{\frac{4}{l-2}, \frac{8}{\sqrt{e}(\sqrt{4r+1}-3)}\}A$. The complete algorithm is given in Fig. 5.

Theorem 14. *The performance ratio of Algorithm DiagonalAlign is* $\rho(1 + \frac{2}{t-2-\frac{1}{c}})$.

Proof. The proof follows the proof of Theorem 8 straightforward. □

4.3. *General score schemes*

When aligning biological sequences, there is often considerable disagreement about how to weight matches, mismatches, insertions, deletions and gaps [5,23]. Many score schemes are proposed and some score schemes satisfy the triangle inequality [16]. To our knowledge, all proposed approximation algorithms with guaranteed performance ratios either explicitly or implicitly assume that the score schemes satisfy the triangle inequality [2,4,15,21,22,24]. In [24], score schemes do not have to satisfy triangle inequality. However, since arbitrary number of intermediate sequences (nodes) are allowed to be added between any two sequences assigned to the two ends of an edge in the topology, one can always obtain a reduced score scheme that satisfies the triangle inequality.

There are cases where the topologies are fixed, e.g., tree alignment, SP alignment (a complete graph is assumed) and consensus alignment (a star is assumed) [4,5,15,16,21,22]. In these cases, if the original score scheme does not satisfy the triangle inequality, no reduced score scheme that satisfies the triangle inequality can be obtained. We further show that the proposed algorithms for consensus alignment work for a very general type of score schemes, i.e., $d(i,i) = 0$ and $d_{\max}/d_{\min} \leqslant$ *constant*, where $d_{\max} = \max_{i \neq j} d(i,j)$ is the largest score for two distinct letters in the alphabet and $d_{\min} = \min_{i \neq j} d(i,j)$ is the smallest score for two distinct letters in the alphabet. The assumption here is very general. In fact, in any score scheme for finite alphabet, $d_{\max}/d_{\min} \leqslant$ *constant* if $d_{\min} \neq 0$. The analysis of our algorithm does not depend on triangle inequality. Contrary to our results, the MAX SNP-hardness in [20] assumes that $d_{\max}/d_{\min} = 1/0$.

Our result mainly depends on the following lemma.

Lemma 15. *Let* $L = \{a_1, a_2, \ldots, a_n\}$ *be a set of* n *letters in* Σ *and* $L^k = \{a_{j_1}, a_{j_2}, \ldots, a_{j_k}\} \subseteq L$ *a set of* k *letters, where* $k = 1 - \delta n$ *for some* $0 \leqslant \delta < \frac{1}{4}$. *Let* $L^r = \{a_{i_1}, a_{i_2}, \ldots, a_{i_r}\}$ *be a multi-set of* r *letters randomly independently chosen from* L^k. *Let* $h(a)$, $h^k(a)$ *and* $h^r(a)$ *denote the numbers of occurrences of the letter* a *in* L, L^k *and* L^r, *respectively. Let* a^r *be the letter minimizing* $\sum_{j=1}^{r} d(a^r, a_{i_j})$ *and* a^* *be the letter minimizing* $\sum_{i=1}^{n} d(a^*, a_i)$. *Then*

$$E\left[\sum_{i=1}^{n} d\left(a^r, a_i\right)\right] \leqslant \left(1 + O\left(\delta + \sqrt{\frac{\log r}{r}}\right)\right) \sum_{i=1}^{n} d(a^*, a_i).$$

Proof. Let

$$c_{\text{opt}} = \sum_{a\in\Sigma} d(a^*, a)h(a)$$

and $c_{\text{alg}} = E[\sum_{a\in\Sigma} d(a^r, a)h(a)]$. We want to prove that $c_{\text{alg}} - c_{\text{opt}} \leqslant O(\delta + \sqrt{\frac{\log r}{r}})c_{\text{opt}}$. It is sufficient to prove that

$$c_{\text{alg}} - c_{\text{opt}} \leqslant O\left(\delta + \sqrt{\frac{\log r}{r}}\right) d_{\min}\big(n - h(a^*)\big).$$

Note that,

$$c_{\text{alg}} = \Pr\big(a^r = a^*\big) \times c_{\text{opt}} + \sum_{a\neq a^*} \Pr\big(a^r = a\big) \times \sum_{a\in\Sigma} d\big(a^r, a\big)h(a).$$

So,

$$c_{\text{alg}} - c_{\text{opt}} \leqslant \Pr\big(a^r \neq a^*\big) d_{\max} n. \tag{17}$$

On the other hand, from the definition of a^r, we have

$$c_{\text{opt}} = \sum_{a\in\Sigma} d(a^*, a)h(a) \geqslant \sum_{a\in\Sigma} d(a^*, a)\frac{k}{r}E\big[h^r(a)\big] = \frac{k}{r}E\left[\sum_{a\in\Sigma} d(a^*, a)h^r(a)\right] \geqslant \frac{k}{r}E\left[\sum_{a\in\Sigma} d\big(a^r, a\big)h^r(a)\right].$$

So,

$$c_{\text{alg}} - c_{\text{opt}} \leqslant E\left[\sum_{a\in\Sigma} d\big(a^r, a\big)\left(h(a) - \frac{k}{r}h^r(a)\right)\right] \leqslant d_{\max} E\left[\sum_{a\in\Sigma}\left(h(a) - \frac{k}{r}h^r(a)\right)\right].$$

Combining with Formula (17), to prove the theorem, it is sufficient to prove that

$$\min\left\{E\left[\sum_{a\in\Sigma}\left(h(a) - \frac{k}{r}h^r(a)\right)\right], \Pr\big(a^r \neq a^*\big)n\right\}. \tag{18}$$

Let $\rho = \frac{d_{\min}}{d_{\min}+d_{\max}}$. We prove Formula (18) in two cases.

Case 1. $n - h(a^*) > \frac{\rho}{4}n$.

Since $n - h(a^*) > \frac{\rho}{4}n$, to prove Formula (18), it is sufficient to show that

$$E\left[\sum_{a\in\Sigma}\left(h(a) - \frac{k}{r}h^r(a)\right)\right] \leqslant O\left(\delta + \sqrt{\frac{\log r}{r}}\right)n.$$

It is easy to see that $\sum_{a\in\Sigma}(h(a) - h^k(a)) \leqslant \delta n$. Therefore, we need only to show that

$$E\left[\sum_{a\in\Sigma}\left(h^k(a) - \frac{k}{r}h^r(a)\right)\right] \leqslant O\left(\sqrt{\frac{\log r}{r}}\right)n. \tag{19}$$

For any $a \in \Sigma$, it is easy to see that $h^r(a)$ is the sum of r independently Poisson trial with success probability $p_a = \frac{h^k(a)}{k}$. Thus, $E[h^r(a)] = p_a r = \frac{r}{k}h^k(a)$. Therefore, by Chernoff's bound [10], for any $0 < \epsilon \leqslant 1$,

$$\Pr\left(\frac{r}{k}h^k(a) - h^r(a) > \epsilon r\right) \leqslant \exp\left(-\frac{\epsilon^2 r}{2p_a}\right) \leqslant \exp\left(-\frac{\epsilon^2 r}{2}\right). \tag{20}$$

Moreover, it is easy to see that $\frac{r}{k}h^k(a) - h^r(a) \leqslant r$. Combining with Formula (20), we know that for any $0 < \epsilon \leqslant 1$,

$$\begin{aligned} E\left[\frac{r}{k}h^k(a) - h^r(a)\right] &\leqslant \epsilon r \times \Pr\left(\frac{r}{k}h^k(a) - h^r(a) \leqslant \epsilon r\right) + r \times \Pr\left(\frac{r}{k}h^k(a) - h^r(a) > \epsilon r\right) \\ &\leqslant \epsilon r + r \times \exp\left(-\frac{\epsilon^2 r}{2}\right). \end{aligned}$$

Let $\epsilon = 2\sqrt{\frac{\log r}{r}}$. The above formula becomes

$$E\left[\frac{r}{k}h^k(a) - h^r(a)\right] \leqslant 2r\sqrt{\frac{\log r}{r}} + \frac{1}{r} \leqslant 3r\sqrt{\frac{\log r}{r}}.$$

Thus, we get,

$$E\left[h^k(a) - \frac{k}{r}h^r(a)\right] \leqslant 3k\sqrt{\frac{\log r}{r}} \leqslant 3n\sqrt{\frac{\log r}{r}}.$$

So, we have proved Formula (19), and thus Formula (18).

Case 2. $n - h(a^*) \leqslant \frac{\rho}{4}n$.

If $a^r \neq a^*$, then by the definition of a^r, $\sum_{j=1}^{r} d(a^r, a_{i_j}) \leqslant \sum_{j=1}^{r} d(a^*, a_{i_j})$. So,

$$d_{\min} \times h^r(a^*) \leqslant d_{\min} \times \big(r - h^r\big(a^r\big)\big) \leqslant d_{\max} \times \big(r - h^r(a^*)\big).$$

Therefore, $(d_{\min} + d_{\max})h^r(a^*) \leqslant d_{\max} r$. By the definition of ρ, we have $r - h^r(a^*) \geqslant \rho r$. Therefore, we have proved that $a^r \neq a^*$ implies $r - h^r(a^*) \geqslant \rho r$. Thus, we can conclude that

$$\Pr\big(a^r \neq a^*\big) \leqslant \Pr\big(r - h^r(a^*) \geqslant \rho r\big). \tag{21}$$

Let $\lambda = \frac{k - h^k(a^*)}{k}$. Then it is easy to verify that

$$\lambda \leqslant \frac{n - h(a^*)}{k} = \frac{n - h(a^*)}{(1-\delta)n} \tag{22}$$

$$\leqslant \frac{\rho}{4(1-\delta)} \leqslant \frac{\rho}{3}. \tag{23}$$

It is easy to see that $r - h^r(a^*)$ is the sum of r independent Poisson trials with success possibility λ. Therefore, by Chernoff's bound [10, Exercise 4.1] and Formula (23), we have

$$\Pr\big(r - h^r(a^*) \geqslant \rho r\big) = \Pr\left(r - h^r(a^*) \geqslant \left(1 + \frac{\rho - \lambda}{\lambda}\right)\lambda r\right) \leqslant \left[e \Big/ \left(1 + \frac{\rho - \lambda}{\lambda}\right)\right]^{\rho r} = (\lambda e/\rho)^{\rho r}.$$

Combining with Formulas (21) and (22), we know that

$$\begin{aligned}
\Pr\big(a^r \neq a^*\big) \times n &\leqslant \frac{n - h(a^*)}{(1-\delta)\lambda} \times \left(\frac{\lambda e}{\rho}\right)^{\rho r} = \frac{1}{1-\delta} \times \frac{e}{\rho} \times \left(\frac{\lambda e}{\rho}\right)^{\rho r - 1} \times \big(n - h(a^*)\big) \\
&\leqslant \frac{1}{1-\delta} \times \frac{e}{\rho} \times \left(\frac{e}{3}\right)^{\rho r - 1} \times \big(n - h(a^*)\big) \qquad (24) \\
&= O\left(\frac{1}{r}\right) \times \big(n - h(a^*)\big), \qquad (25)
\end{aligned}$$

where Inequality (24) is from Formula (23). Therefore, we have proved Formula (18), and thus the lemma follows. □

With Lemma 15, Algorithms AverageConsensusAlign and DiagonalAlign can be easily extended (instead of finding *majority* sequence in step 1(a) of AverageConsensusAlign, find the sequence formed by those a^r's as described in Lemma 15) to PTAS for the general score schemes.

5. Concluding remarks

If we look at the ratios of our algorithms for AVERAGE c-GAP CONSENSUS ALIGNMENT and AVERAGE c-GAP SP ALIGNMENT, we may notice an interesting phenomenon that the later ratio is better than the former. More specifically, the ratio for the consensus model is $O(\frac{1}{l} + \frac{1}{\sqrt{r}})$ and for the SP model is $O(\frac{1}{l} + \frac{1}{r})$. This is somewhat surprising. Our algorithms approximate a median sequence and a frequency matrix for the consensus model and

the SP model, respectively. How can we hope that the approximation of a frequency matrix is easier than that of a sequence? We can explain this phenomenon roughly as follows: The cost function is linear for the consensus model and is quadratic for the SP model. If we can approximate a linear function with an $O(\frac{1}{\sqrt{r}})$ ratio, then we are hopeful to approximate the quadratic function with an $O(\frac{1}{r})$ ratio.

We have also defined the problem c-GAP SP ALIGNMENT in Section 1 and said that it is an easier version of AVERAGE c-GAP SP ALIGNMENT. In fact, we can give a PTAS for this version by setting $l = 1$ in Algorithm AverageSPAlign. Obviously, the ratio of the above algorithm is $O(1 + \frac{2}{r})$, which can be proved by setting $\delta = 0$ in Theorem 3.

In Algorithms AverageSPAlign and AverageConsensusAlign, when using as subroutines of the algorithms for c-diagonal models, the **for** statement in step 1 can be replaced by the following statement as below and the running time is significantly reduced.

1. **for** L from m to $(2c + 1)m$ **do**
   for any $s_{i_1}, s_{i_2}, \ldots, s_{i_r} \in \mathcal{S}$ **do**
   for any possible alignment $\mathcal{M}'$ of $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ such that the length is L and each sequence contains no more than cl insertions and deletions **do** ...

Acknowledgments

We would like to thank Tao Jiang for helpful discussions and Annie Lee for comments and corrections. Bin Ma is supported by NSERC, PREA, the Canada Research Chairs Program, and NSF of China 60553001. He was supported by HK RGC Grants 9040297, 9040352, and CityU Strategic Grant 7000693 while visiting CityU. Lusheng Wang is supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 120905]. Ming Li was supported in part by City University of Hong Kong, the NSERC Research Grant OGP0046506, Canada Research Chair program, and the Steacie Fellowship.

References

[1] S. Altschul, D. Lipman, Trees, stars, and multiple sequence alignment, SIAM J. Appl. Math. 49 (1989) 197–209.
[2] V. Bafna, E. Lawler, P. Pevzner, Approximation algorithms for multiple sequence alignment, in: Proc. 8th Ann. Combinatorial Pattern Matching Conf. Asilomar, 1994, pp. 43–53.
[3] K. Chao, W.R. Pearson, W. Miller, Aligning two sequences within a specified diagonal band, CABIOS 8 (1992) 481–487.
[4] D. Gusfield, Efficient methods for multiple sequence alignment with guaranteed error bounds, Bull. Math. Biol. 30 (1993) 141–154.
[5] D. Gusfield, Algorithms on Strings, Trees, and Sequences, Cambridge University Press, 1997.
[6] J.W. Fickett, Fast optimal alignment, Nucleic Acids Res. 12 (1984) 175–180.
[7] W. Just, On the computational complexity of gap-0 multiple alignment, manuscript, 1998.
[8] J. Kececioglu, H.-P. Lenhof, K. Mehlhorn, P. Mutzel, K. Reinert, M. Vingron, A polyhedral approach to sequence alignment problems, in: P. Pevzner (Ed.), Special Issue on Computational Biology, Discrete Appl. Math. (1999), in press.
[9] M. Li, B. Ma, L. Wang, Finding similar regions in many sequences, in: Proc. 31st ACM Symp. Theory of Computing, Atlanta, 1999, pp. 473–482.
[10] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
[11] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.
[12] W.R. Pearson, D. Lipman, Improved tools for biological sequence comparison, Proc. Natl. Acad. Sci. USA 85 (1988) 2444–2448.
[13] W.R. Pearson, Rapid and sensitive sequence comparison with FASTP and FASTA, Methods Enzymol. 183 (1990) 63–98.
[14] W.R. Pearson, Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith–Waterman and FASTA algorithms, Genomics 11 (1991) 635–650.
[15] P. Pevzner, Multiple alignment, communication cost, and graph matching, SIAM J. Appl. Math. 52 (1992) 1763–1779.
[16] D. Sankoff, J. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Addison–Wesley, 1983.
[17] J.L. Spouge, Fast optimal alignment, CABIOS 7 (1991) 1–7.
[18] C.J. Stone, A Course in Probability and Statistics, Duxbury Press, 1995.
[19] E. Ukkonen, Algorithms for approximate string matching, Inform. Control 64 (1985) 100–118.
[20] L. Wang, T. Jiang, On the complexity of multiple sequence alignment, J. Comput. Biol. 1 (1994) 337–348.
[21] L. Wang, T. Jiang, E.L. Lawler, Approximation algorithms for tree alignment with a given phylogeny, Algorithmica 16 (1996) 302–315.
[22] L. Wang, D. Gusfield, Improved approximation algorithms for tree alignment, J. Algorithms 25 (1997) 255–273.
[23] M.S. Waterman, Introduction to Computational Biology, Chapman & Hall, 1995.
[24] B.Y. Wu, G. Lancia, V. Bafna, K. Chao, R. Ravi, C.Y. Tang, A polynomial time approximation scheme for minimum routing cost spanning trees, in: Proc. 9th ACM–SIAM Symp. Disc. Alg., San Francisco, 1998, pp. 21–32.