Near Optimal Multiple Alignment Within a Band In Polynomial Time

Ming Li* University of Waterloo

Bin Ma† University of Waterloo

Lusheng Wang‡ City University of Hong Kong

ABSTRACT

Multiple sequence alignment is one of the most important problems in computational biology. Because of its notorious difficulties, aligning sequences within a constant band is a popular practice in bioinformatics with good results [17; 13; 14; 15; 1; 3; 6; 20; 18]. However, the problem is still NP-hard for multiple sequences. In this paper, we present polynomial time approximation schemes (PTAS) for multiple sequence alignment within a constant band, under standard models of SP alignment and consensus (star) alignment. The algorithms work for very general score schemes.

In order to prove our main results, we also present a PTAS for SP alignment and a PTAS for consensus alignment, allowing only constant number of insertion and deletion gaps (of arbitrary length) per sequence on the average.

1. INTRODUCTION

Multiple sequence alignment is one of the fundamental and most challenging problems in computational molecular biology [1; 17; 4; 5; 24]. It plays an essential role in finding similarity and highly conserved subregions among a set of biological sequences.

Many objective functions have been proposed for measuring the quality of the alignments. We only consider the commonly used standard models [5; 24]: the *SP alignment* and the *Consensus alignment* (also called Star-alignment). Given sequences $s_1, \dots, s_n$ over alphabet Σ, let $\mathcal{M}$ be a multiple alignment of them. 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 j-th letter of S_i by $S_i[j]$. The majority sequence of $S_1, S_2, \dots, S_n$ is a sequence S over $\Sigma \cup \{\Delta\}$ such that $S[j]$ is the majority letter of $S_1[j], S_2[j], \dots, S_n[j]$.

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 $S_1, S_2, \dots, S_n$ 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 [21]. 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 [4; 16; 2]. The $2 - o(1)$ barrier appears to be formidable. There is an enormous literature and various other models, methods, and heuristics on multiple sequence alignment for which we refer the reader to [5; 24], and [8] for some recent developments. In this paper, we are interested in theoretically resolving a popular special case of the problem: *multiple alignment within a band.* The restriction of aligning within c-diagonal band is often applied in many practical cases. Methods under this assumption have been extensively studied. Sankoff and Kruskal discussed the problem under the rubric of "cutting corners" in [17]. Alignment within a band is used in the final stage of the FASTA program for rapid searching of protein and DNA sequence databases [13; 14]. Pearson has shown that alignment within a band gives very good results for lots of protein superfamilies [15]. Other references can be found in [1; 3; 6; 20]. Spouge gives a survey on this topic in [18]. We now define the problem.

Definition 3. c-DIAGONAL ALIGNMENT Let $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ be a set of n sequences, each of length m, and $\mathcal{M}$ an

---

*Supported in part by the NSERC Research Grant OGP0046506, a CGAT grant, CITO, the Steacie Fellowship. Address: Dept. Comput. Sci., Univ. of Waterloo, Waterloo, Ont. N2L 3G1, Canada. E-mail: mli@math.uwaterloo.ca

†Supported in part by HK RGC Grants 9040297, 9040352, and CityU Strategic Grant 7000693 while visiting CityU and by NSERC Research Grant OGP0046506. This work was partly done when in Peking University. Address: Dept. Comput. Sci., Univ. of Waterloo, Waterloo, Ont. N2L 3G1, Canada. E-mail: b3ma@wh.math.uwaterloo.ca

‡Supported in part by HK RGC Grants 9040297, 9040352, 9040444 and CityU Strategic Grant 7000693. Address: Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: lwang@cs.cityu.edu.hk

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

Figure 1: One insertion gap corresponds to three deletion gaps.

alignment of the n sequences. 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 p-th letter of s_i is in column p' of $\mathcal{M}$, and the p-th letter of s_j is in column q' of $\mathcal{M}$, then $|p'-q'| \leq c$. In other words, the inserted spaces are "evenly" distributed among all sequences and the i-th position of a sequence is about at most c positions away from the i-th position of any other sequences.

c-DIAGONAL ALIGNMENT remains to be NP-hard in both 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 for the SP alignment model in Section 2, Corollary 1. The main results of the paper are PTAS's for c-DIAGONAL ALIGNMENT under both models. The algorithms also work for a relaxed version of c-Diagonal Alignment, where we know the "approximate" positions of all letters in all sequences in an optimal alignment. For example, the j-th letter of s_i is at most c positions away from the k-th letter of s_j in the alignment, where c is a constant. In order to obtain our main results, we also need to design PTAS's 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 Figure 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 a PTAS 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 frequence of a letter a in the r letters is very close to its frequence 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. Then 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. Since 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. (Sketch) W. Just has recently 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 whose every node has degree no more than 3. It is known to be Max SNP-hard [12] and hence NP-hard. Let $G = <V, E>$ 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 $\Sigma = \{0, 1, x\}$. 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:

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

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

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

Note that $\texttt{X} = \texttt{x}^M$ and $p_{k,j} = \texttt{XxxxxxX}$ 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 $1 \le i \le n$, 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 $N \times ((m-K)\times(12n-18)+K\times(12n-20)) + 4\times|V_1|\times|V_2| = Nm(12n-18) - 2NK + 4\times|V_1|\times|V_2| \le Nm(12n-18) - 2NK + N$. 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. $\square$

In the above proof, since $\mathtt{X} = \mathtt{x}^M$, M is large and $\deg(v) \le 3$ 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 1. *For any $c \ge 0$,* 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 j-th position of $\mathcal{M}$. The SP-score of $\mathcal{M}$ can be rewritten as:

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

Clearly, $\frac{x_{j,a}}{n}$ is the frequency of letter a in the j-th 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 Figure 2.

Theorem 2. *If $l \ge 4$, $r \ge 1$, algorithm AverageSPAlign in Figure 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.

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 j-th position of the alignment $\mathcal{M}'$ for $j = 1, 2, \ldots, n$ 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).

Figure 2: A PTAS for Average c-Gap SP Alignment.

Proof. Let $\mathcal{M}_{opt}$ be an optimal Average c-Gap SP Alignment of $s_1, s_2, \ldots, s_n$ and L be the length of $\mathcal{M}_{opt}$. Let S'_i be the supersequence of s_i in $\mathcal{M}_{opt}$. For any $l > 0$, since the total number of gaps is no more than cn in $\mathcal{M}_{opt}$, 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 $\mathcal{M}_{opt}$ 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)(b)(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 j-th position of $\mathcal{M}_{opt}$, and $y_{j,a}$ be the number of the occurrences of letter a at the j-th position of $\mathcal{M}$. Then to prove the theorem, we only need to prove that

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

We prove Inequality 1 via several claims.

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

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

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

PROOF. It is sufficient to show that

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

This follows from Claim 1 and $\sum_{a\in\Sigma\cup\{\Delta\}}(y_{j,a}-\tilde{x}_{j,a})=0$. □

CLAIM 3. $\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}y_{j,b}\leq\sum_{j=1}^{L}\sum_{a\neq b}\tilde{x}_{j,a}x_{j,b}$.

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

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

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 $\sum_{j=1}^{L}\lambda_{j,S_i[j]}\geq\sum_{j=1}^{L}\lambda_{j,S'_i[j]}$. Therefore, the claim is proved. □

CLAIM 4.

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

PROOF. It is sufficient to show that

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

Since $\sum_{a\in\Sigma\cup\{\Delta\}}(\tilde{x}_{j,a}-x_{j,a})=0$, by Claim 1, 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

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

Let S'_i be the supersequence of $s_i\in\mathcal{S}'$ for $i=1,2,\ldots,(1-\delta)n$ in $\mathcal{M}_{opt}$. 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., $\lambda_{j,a}$ satisfies a binomial distribution $B(r,\frac{x''_{j,a}}{n})$. By a simple property of binomial distribution [19],

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

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

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

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

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

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 $x_{j,a}\leq x''_{j,a}$, then we have

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

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

Case 2: If $x_{j,a}>x_{j,a}$, we still have

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

Therefore, we have

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

Secondly,

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

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

$$x''_{j,a}(n - x''_{j,a}) \leq \frac{1}{(1-\delta)^2} x_{j,a}(n - x_{j,a}) \quad (8)$$

Combining formulas (6)(7)(8), we know that

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

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

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

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

Combining claims 2, 3 and 4, Formula (1) is proved, hence the theorem. □

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 Figure 3.

Theorem 3. *The performance ratio of algorithm diagonalSPAlign is* $\rho(1 + \frac{2}{t-2-\frac{1}{c}})$.

Proof. Let $\mathcal{M}_{opt}$ 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 $c_{opt}(i)$ 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 $c_{alg}(i)$ be the alignment cost from algorithm AverageSPAlign. Then we have

Claim 5. $c_{alg}(i) \leq \rho c_{opt}(i)$.

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 $c_i \leq \rho ctn^2$. 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.	Put the alignments of the segments together to form a multiple alignment for the original $\mathcal{S}$.

Figure 3: A PTAS for c-Diagonal Alignment in SP model.

Proof. If $c_{opt}(i) \geq ctn^2$, then from the selection of L in step 2, $c_{alg}(i) \leq \rho ctn^2$. So we have $c_{alg}(i) \leq \rho c_{opt}(i)$. If $c_{opt}(i) < ctn^2$, 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 2, we also have $c_{alg}(i) \leq \rho c_{opt}(i)$. □

If $h = 1$, the theorem is trivially true by Claim 5. So, in the rest of the proof, we assume $h \geq 2$. Let $c(i)$ be the cost in $\mathcal{M}_{opt}$ contributed by the i-th 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 j-th position of $\mathcal{M}_{opt}$. Then we have

Claim 6. *For any* $1 \leq i < h$, $c(i) \geq c(t-2)n^2 - n^2$.

Proof. We prove it by contradiction. Suppose that $c(i) < c(t-2)n^2 - n^2$ for some $1 \leq i < h$. Then since the c-diagonal condition, the alignment of the i-th segment which is induced by $\mathcal{M}_{opt}$ has score less than $c(i) + 2cn^2$. Therefore, $c_{opt}(i) < ctn^2 - n^2$. 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 $c_{opt}(i) + n^2 < ctn^2$. Thus, by Theorem 2, 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 6, we have

$$c_{opt} \geq \sum_{i=1}^{h-1} c(i) \geq (h-1)(c(t-2)n^2 - n^2). \quad (9)$$

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

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

Combining with Claim 5, we have

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

Algorithm averageConsensusAlign

Input: integers c, l, r, and $\mathcal{S} = \{s_1, \dots, s_n\}$, each s_i has length m.

Output: a multiple alignment $\mathcal{M}$.

1. **for** L from m to nm **do**
   for any $s_{i_1}, s_{i_2}, \dots, s_{i_r} \in \mathcal{S}$ **do**
   for any possible alignment $\mathcal{M}'$ of $s_{i_1}, s_{i_2}, \dots, s_{i_r}$ 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 super-sequences of s_{i_1}, s_{i_2} $\dots$, 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 $\mathcal{M} = \{S_1, S_2, \dots, S_n\}$. 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 minized among all $\mathcal{M}$'s obtained in step 1(c).

Figure 4: PTAS for AVERAGE c-GAP CONSENSUS ALIGNMENT.

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

$$\begin{aligned}\frac{\sum_{i=1}^{h} c_{alg}(i)}{c_{opt}} &\leq \rho + \frac{2\rho(h-1)cn^2}{(h-1)(c(t-2)n^2 - n^2)} \\ &\leq \rho(1 + \frac{2}{t-2-\frac{1}{c}}). \quad \Box\end{aligned}$$

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's 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 2). Using this stronger property, we design the PTAS for average c-gap consensus alignment.

THEOREM 4. *For $l > 2$, $r > 2$, algorithm averageConsensusAglign gives an alignment with cost at most*

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

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

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

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

From Lemma 1, we want to prove the following lemma:

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

$$\begin{aligned}&k^{-r} \sum_{i_1, i_2, \dots, i_r \in J} [h(a^*) - h(a_{(i_1, i_2, \dots, i_r)})] \\ \leq\ &\max\{\frac{4}{l-2}, \frac{8}{\sqrt{e}\,(\sqrt{4\,r+1}-3)}\}(A-1)(n-h(a^*)).\end{aligned} \tag{11}$$

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

$$J^r = \{(i_1, i_2, \dots, i_r) | i_j \in J\}$$

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

$$\begin{aligned}\mathcal{I}_a &= \{(i_1, i_2, \dots, i_r) \in J^r | a \text{ is a majority of } a_{i_1}, a_{i_2}, \dots, a_{i_r}\} \\ \mathcal{L}_a &= \{(l_1, l_2, \dots, l_A) | l_1 + l_2 + \dots + l_A = r \text{ and } l_b \leq l_a \text{ for any } b \in \Sigma\}.\end{aligned}$$

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

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

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

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

Two cases arise:

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

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

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

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

Let x_a be the number of occurrences of letter a in $A_J = \{a_{i_1}, a_{i_2}, \dots, a_{i_k}\}$. Then for any $a \in \Sigma$, $x_{a^*} \geq h(a^*) - n/l \geq h(a^*) - 2n/l \geq h(a) \geq x_a$. 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

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

For any $(l_1, l_2, \dots, l_A) \in \mathcal{L}_a$, since $l_{a^*} \leq l_a$, and $x_{a^*} \geq x_a$, we have $x_a^{l_a} x_{a^*}^{l_{a^*}} \leq (\sqrt{x_a x_{a^*}})^{l_a} (\sqrt{x_a x_{a^*}})^{l_{a^*}}$. 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

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

Combining with Formula (14) and Lemma 1,

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

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

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

Combining with Formula (15), we have

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

So,

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

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

Lemma 2 actually shows that $h(a_{i_1, i_2, \dots, i_r})$ is very close to $h(a^*)$ in expection. 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 $S_{j_1}, S_{j_2}, \dots, S_{j_k}$ as the majority sequence of $S_1, S_2, \dots, S_n$, 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 $|J| \geq (1 - 1/l)n$. From Lemma 2, we know that the performance ratio of the algorithm averageConsensusAlign is

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

Algorithm diagonalAlign	
Input:	integers c, l, r and t, and $\mathcal{S} = \{s_1, \dots, s_n\}$, 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 $\{s_1[1..i], s_2[1..i], \dots s_n[1..i]\}$.
2.	Let L be the maximum i such that $c_i \leq \rho ctn$. 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 $\mathcal{S}$.

Figure 5: A PTAS for c-Diagonal Alignment in consensus model

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

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 Figure 5.

Theorem 5. *The performance ratio of algorithm diagonalAlign is* $\rho(1 + \frac{2}{t-2-\frac{1}{c}})$.

Proof. The proof follows the proof of Theorem 3 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; 24]. Many score schemes are proposed and some score schemes satisfy the triangle inequality [17]. To our knowledge, all proposed approximation algorithms with guaranteed performance ratios either explicitly or implicitly assume that the score schemes satisfy the triangle inequality [4; 16; 2; 22; 23; 25]. In [25], 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; 17; 16; 22; 23; 5]. 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 $d_{max}/d_{min} \leq constant$, 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, $d_{max}/d_{min} \leq constant$ 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 [21] assumes that $d_{max}/d_{min}=1/0$. In this section, we show that algorithm diagonalAlign works for a very general type of score scheme, i.e. $d(i,i)=0$ and d_{max}/d_{min} is bounded by a constant, where $d(i,j)$ is the score between the pair of letters i and j, $d_{max}=\max_{i\neq j} d(i,j)$, and $d_{min}=\min_{i\neq j} d(i,j)$. Our result mainly depends on the following lemma.

LEMMA 3. *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 $0\leq\delta<\frac{1}{4}$. 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*

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

PROOF. Let

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

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

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

Note that,

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

So,

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

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

$$\begin{aligned} c_{opt} &= \sum_{a\in\Sigma} d(a^*,a)h(a)\\ &\geq \sum_{a\in\Sigma} d(a^*,a)\frac{k}{r}E[h^r(a)]\\ &= \frac{k}{r}E[\sum_{a\in\Sigma} d(a^*,a)h^r(a)]\\ &\geq \frac{k}{r}E[\sum_{a\in\Sigma} d(a^r,a)h^r(a)]. \end{aligned}$$

So,

$$\begin{aligned} c_{alg}-c_{opt} &\leq E[\sum_{a\in\Sigma} d(a^r,a)\left(h(a)-\frac{k}{r}h^r(a)\right)]\\ &\leq d_{max}E[\sum_{a\in\Sigma}\left(h(a)-\frac{k}{r}h^r(a)\right)]. \end{aligned}$$

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

$$\begin{aligned} &\min\{E[\sum_{a\in\Sigma}\left(h(a)-\frac{k}{r}h^r(a)\right)],\Pr(a^r\neq a^*)n\}\\ &\leq O\left(\delta+\sqrt{\frac{\log r}{r}}\right)(n-h(a^*)). \end{aligned} \tag{18}$$

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

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

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

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

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 $0<\epsilon\leq 1$,

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

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

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

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

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

Thus, we get,

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

So, we have proved Formula (19), and thus Formula (18).
Case 2 $n-h(a^*)\leq\frac{\rho}{4}n$.
If $a^r\neq a^*$, then by the definition of a^r, $\sum_{j=1}^r d(a^r,a_{i_j})\leq\sum_{j=1}^r d(a^*,a_{i_j})$. So,

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

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

$$\Pr(a^r \ne a^*) \le \Pr(r - h^r(a^*) \ge \rho r). \tag{21}$$

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

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

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

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

$$\begin{aligned}\Pr\left(r - h^r(a^*) \ge \rho r\right) &= \Pr\left(r - h^r(a^*) \ge (1+\frac{\rho-\lambda}{\lambda})\lambda r\right) \\ &\le \left[e/(1+\frac{\rho-\lambda}{\lambda})\right]^{\rho r} = (\lambda e/\rho)^{\rho r}.\end{aligned}$$

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

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

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

With Lemma 3, 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 3) to PTAS's 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? Actually, 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 $O(\frac{1}{\sqrt{r}})$ ratio, then we are hopeful to approximate the quadratic function with $O(\frac{1}{r})$ ratio.

The PTAS's for both SP model and consensus model can be extended to work for the case where we know the "approximate" positions of all letters in all sequences in an optimal alignment. That is, the j-th letter of s_i is at the position that is at most c positions away (either left or right) from the k-th letter of another sequence, where c is a constant. The difference here is that in an optimal alignment, some sequences have lots of spaces in a segment, whereas some sequences are very similar. In other words, the spaces are not evenly distributed for some sequences. The algorithm is the same except that we may cut the segment at different locations.

THEOREM 6. *The relaxed* c-DIAGONAL ALIGNMENT *has a PTAS, in both SP model and the consensus model.*

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 AvereageSPAlign. Obviously, the ratio of the above algorithm is $O(1+\frac{2}{r})$, which can be proved by setting $\delta = 0$ in Theorem 2.

In Algorithm 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 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**

Acknowledgements. We would like to thank Tao Jiang for helpful discussions.

6. REFERENCES

[1] S. Altschul and D. Lipman, Trees, stars, and multiple sequence alignment, *SIAM J. Appl. Math.* 49, pp. 197-209, 1989.

[2] V. Bafna, E. Lawler and P. Pevzner, Approximation algorithms for multiple sequence alignment, *Proc. 8th Ann. Combinatorial Pattern Matching Conf.*, pp. 43-53, 1994.

[3] K. Chao, W. R. Pearson and W. Miller, Aligning two sequences within a specified diagonal band, *CABIOS*, 8, pp. 481-487, 1992.

[4] D. Gusfield, Efficient methods for multiple sequence alignment with guaranteed error bounds, *Bull. Math. Biol.*, 30, pp. 141-154, 1993.

[5] D. Gusfield, *Algorithms on strings, trees, and sequences*, Cambridge Univ. Press, 1997.

[6] J. W. Fickett, Fast optimal alignment, *Nucleic Acids Res.*, 12, 175-180, 1984.

[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. To appear in *Disc. Appl. Math.*, special issue on Computational Biology, Ed. P. Pevzner, 1999.

[9] M. Li, B. Ma, and L. Wang, Finding similar regions in many sequences, *Proc. 31st ACM Symp. Theory of Computing*, 473-482, Atlanta, 1999.

[10] R. Motwani and P. Raghavan, *Randomized algorithms*, Cambridge University Press, 1995.

[11] G. Myers, S. Selznick, Z. Zhang, W. Miller, Progressive multiple alignment with constraints, *1st Ann. Int'l Conf. Comput. Mol. Biol.*, pp. 220-225, 1997.

[12] C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *J. Comput. System Sci.*, 43, 425-440, 1991

[13] W. R. Pearson and D. Lipman, Improved tools for biological sequence comparison, *Proc. Natl. Acad. Sci. USA*, 85, pp. 2444-2448, 1988.

[14] W. R. Pearson, Rapid and sensitivecomparison with FASTP and FASTA, *Methods Enzymol.*, 183, pp. 63-98, 1990.

[15] W. R. Pearson, Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms, *Genomics*, 11, pp. 635-650, 1991.

[16] P. Pevzner, Multiple alignment, communication cost, and graph matching, *SIAM J. Appl. Math.*, 52, pp. 1763-1779, 1992.

[17] D. Sankoff and J. Kruskal, *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison,* Addison Wesley, 1983.

[18] J. L. Spouge, Fast optimal alignment, *CABIOS*, 7, pp. 1-7, 1991.

[19] C. J. Stone, *A Course in Probability and Statistics*, Duxbury Press, 1995.

[20] E. Ukkonen, Algorithms for approximate string matching, *Inform. Control*, 64, pp. 100-118, 1985.

[21] L. Wang and T. Jiang, On the complexity of multiple sequence alignment, *J. Comp. Biol.*, 1, pp. 337-348, 1994.

[22] L. Wang, T. Jiang and E. L. Lawler, Approximation algorithms for tree alignment with a given phylogeny, *Algorithmica*, 16, pp. 302-315, 1996.

[23] L. Wang and D. Gusfield, Improved approximation algorithms for tree alignment, *J. Alg.*, 25, pp. 255-273, 1997.

[24] M. S. Waterman, *Introduction to computational biology*, Chapman & Hall, 1995.

[25] 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, *Proc. 9th ACM-SIAM Symp. Disc. Alg.*, pp. 21-32, 1998.