COMPUTATIONAL COMPLEXITY OF MULTIPLE SEQUENCE ALIGNMENT WITH SP-SCORE

WINFRIED JUST
DEPARTMENT OF MATHEMATICS
OHIO UNIVERSITY
ATHENS, OHIO 45701, U.S.A.

ABSTRACT. It is shown that the multiple alignment problem with SP-score is $\mathcal{NP}$-hard for each scoring matrix in a broad class $\mathcal{M}$ that includes most scoring matrices actually used in biological applications. The problem remains $\mathcal{NP}$-hard even if sequences can only be shifted relative to each other and no internal gaps are allowed.

It is also shown that there is a scoring matrix M_0 such that the multiple alignment problem for M_0 is MAX-$\mathcal{SNP}$-hard, regardless of whether or not internal gaps are allowed.

Key words and phrases. sequence alignment, scoring matrix, $\mathcal{NP}$-hardness, MAX-$\mathcal{SNP}$-hardness, polynomial time approximation scheme.

e-mail: just@math.ohiou.edu phone: (740)-593-1260 fax: (740)-593-9805.

1. Introduction

The importance of good multiple sequence alignment algorithms is evidenced by the large number of programs that have been developed for this task (Fasman and Salzberg 1998). Finding an optimal alignment of k sequences appears to quickly become computationally intractable as k increases. For example, dynamic programming algorithms that are guaranteed to find a best scoring alignment of k sequences with mean length n have a running time of $O(n^k)$ (Carillo and Lipman 1988). This explains the widespread use of heuristic algorithms for multiple alignment. It has been formally proved by Wang and Jiang (1994) and Bonizzoni and Della Vedova (2000) that there are scoring matrices for which the problem of finding a multiple alignment of k sequences with optimal SP-score is $\mathcal{NP}$-hard. Unfortunately, the scoring matrix used by Wang and Jiang (1994) for obtaining this result is not a metric, which makes it very different from the matrices that are actually used in biological applications. The proof technique used by Bonizzoni and Della Vedova (2000) uses matrices in which the indel (insertion/deletion) penalties depend on which character a space symbol is aligned with. While such variable indel penalties are sometimes used for aligning amino acid sequences, the use of scoring schemes with uniform indel penalties seems much more common. Thus for most scoring schemes used in practice, computational intractability of the multiple alignment problem had not been formally proven prior to the results of the present paper. Here we show that the multiple alignment problem is $\mathcal{NP}$-hard for each scoring matrix from a broad class $\mathcal{M}$ that includes most scoring schemes that are actually used in biological applications.

A brute force algorithm for finding optimal multiple alignments would have to evaluate all possibilities of inserting gaps into the sequences to be aligned. However, the optimal alignments found in practice usually contain relatively few gaps (Pascarella and Argos 1992), (Benner *et al.* 1993). This observation led to the question whether the problem becomes less complex if one limits the number of gaps that can be inserted into the sequences (Jiang 1999). An extreme version of such restrictions is what we call here *gap-0 alignment.* In this version, sequences can be shifted relative to each other, but no internal gaps are allowed. Unpublished results of Bonizzoni, Della Vedova, and Jiang show that there is a scoring matrix that does not satisfy the triangle inequality for which gap-0 alignment is still $\mathcal{NP}$-hard, and the problem is even MAX-$\mathcal{SNP}$-hard if the scoring matrix is considered part of the input (Jiang 1999). Subsequently, a fixed scoring matrix M was found such that M is a metric and gap-0 multiple alignment for M is $\mathcal{NP}$-hard (Just 1999). Here we show that the gap-0 multiple alignment problem is $\mathcal{NP}$-hard for each scoring matrix from a broad class $\mathcal{M}_1 \supset \mathcal{M}$. We also show that there is a fixed scoring matrix M_0 over a three-letter alphabet such that the multiple alignment problem and the gap-0 multiple alignment problem for M_0 are MAX-$\mathcal{SNP}$-hard. Unfortunately, M_0 does not satisfy the triangle inequality.

2. Definitions and Results

Let us formally state the multiple alignment problem and the gap-0 multiple alignment problem. At the outset, we are given a finite alphabet $\Sigma = \{a_1, ..., a_w\}$ and a $(w+1) \times (w+1)$ scoring matrix $M = (s_{i,j})_{i \leqslant w, j \leqslant w}$. Intuitively, for $i, j > 0$, $s_{i,j}$ represents the penalty for aligning character a_i with character a_j. For $i > 0$, the numbers $s_{0,i}, s_{i,0}$ are called *indel penalties.* Penalties $s_{0,i}, s_{i,0}$ are incurred whenever

the character a_i is aligned with a special character $\Delta \notin \Sigma$ that stands for a space. A given scoring scheme may also specify additional *gap opening penalties* that are incurred in addition to the indel penalties for aligning a_i with the first or last Δ in a string of Δ's (in this case, what we call "indel penalty" will usually be called *gap extension penalty*). Our results do not depend on whether or not gap opening penalties are added to the indel penalties.

We will say that a scoring matrix is *metric* if it satisfies the following conditions:
1) $s_{i,j} > 0$ for all $i \neq j$;
2) $s_{i,i} = 0$ for all i;
3) $s_{i,j} = s_{j,i}$ for all i, j.
4) $s_{i,j} + s_{j,k} \geqslant s_{i,k}$ for all i, j, k.
The last of the above properties is called the *triangle inequality.*

Metric scoring matrices are of considerable theoretical interest, since they allow for the natural interpretation of pairwise alignment scores as distances between sequences (see e.g. (Wheeler 1993) and (Fitch 1993) for a discussion of the role of the triangle inequality in this context). However, scoring matrices used in practice, such as the PAM matrices of Dayhoff *et al.* (1978) and the BLOSUM matrices of Henikoff and Henikoff (1992) give log-odds scores rather than distances. In particular, for the latter type of matrices, the multiple alignment problem will be formally cast as a maximization rather than a minimization problem. In this paper we will use the language of "distances" as a convenient and intuitive metaphor, but our development of the theory and our results will not require any of the properties 1)-4). A maximization problem can of course be transformed into an equivalent minimization problem by multiplying each score by -1.

Given two sequences t_0, t_1 of symbols from $\Sigma \cup \{\Delta\}$ of length n and a scoring matrix M, we define a distance $d_M(t_0, t_1)$ as the sum of penalties specified by M for aligning the j-th character $t_{0,j}$ of t_0 with the j-th character $t_{1,j}$ of t_1, plus gap opening penalties if applicable, where j ranges over the length of the sequences. If we have a k-tuple $\langle t_0, ..., t_{k-1} \rangle$ of sequences of equal length, then the *SP-score* for these sequences is given by $SP^M(t_0, ..., t_{k-1}) = \sum_{i<j<k} d_M(t_i, t_j)$.

For a k-tuple $\langle t_0, ..., t_{k-1} \rangle$ of sequences as above, an *alignment* a of these sequences is obtained by preserving the order of symbols in each sequence, but possibly inserting space symbols Δ. We will always assume that there are suitable numbers of space symbols inserted at the end of each sequence so that the aligned sequences $\langle at_0, \ldots, at_{k-1} \rangle$ are all of the same length. Alignments are not allowed to contain columns that consist entirely of space symbols. An alignment a is called a *gap-0 alignment* if spaces are possibly added at the beginning and at the end of sequences, but not between symbols (i.e., sequences can be shifted relative to each other, but no internal gaps are allowed). A *gap-0-1 alignment* is a gap-0 alignment of sequences of equal length such that each of the aligned sequences contains exactly one space, either at its end or at its beginning.

Given an alignment a of sequences $\langle t_0, ..., t_{k-1} \rangle$, we define the *SP-score with respect to M* for this alignment as $SP^M(at_0, ..., at_k)$. Now let us formally define the *multiple alignment problem,* the *gap-0 multiple alignment problem,* and the *gap-0-1 multiple alignment problem* for a given alphabet Σ and scoring matrix M. In each case, the instance is a k-tuple of sequences of common length[1] of characters

[1] In most biological applications, the sequences to be aligned have approximately equal length, but not necessarily exactly equal length. Note that if multiple alignment of sequences of exactly

from Σ. The problem is to find a multiple alignment (respectively gap-0 multiple alignment, or gap-0-1 multiple alignment) of the given sequences that minimizes the SP-score with respect to M.

Now let $\Sigma = \{A, T\}$ and let us say that a scoring matrix M is *generic* if it is of the form

	Δ	A	T
Δ	x	y	z
A	y	v_A	u
T	z	u	v_T

FIG. 1. A generic scoring matrix.

where the parameters x, y and z are fixed nonnegative numbers[2] and the inequality $u > \max\{0, v_A, v_T\}$ holds. Let us say that a $(w+1) \times (w+1)$ scoring matrix N *contains a generic submatrix* if there are $1 \leq i, j \leq w$ such that after deleting all rows and colums of N except those numbered $0, i, j$ one obtains a generic matrix M. Now let $\mathcal{M}_2$ be the class of all scoring matrices that contain a generic submatrix M, let $\mathcal{M}_1$ be the class of all scoring matrices that contain a submatrix isomorphic to a generic matrix M with $z > v_T$, and let $\mathcal{M}$ be the class of all scoring matrices that contain a submatrix isomorphic to a generic matrix M with $y > u$ and $z > v_T$.

Recall that an optimization problem is $\mathcal{NP}$*-hard* if the existence of a polynomial-time algorithm that is guaranteed to find the optimal solution for all instances of this problem implies that $\mathcal{P} = \mathcal{NP}$ (Garey and Johnson 1979). Here is the main result of this paper.

Theorem 1. (a) *For every scoring matrix $M \in \mathcal{M}$, the multiple alignment problem is $\mathcal{NP}$-hard.*
(b) *For every scoring matrix $M \in \mathcal{M}_1$, the gap-0 multiple alignment problem is $\mathcal{NP}$-hard.*
(c) *For every scoring matrix $M \in \mathcal{M}_2$, the gap-0-1 multiple alignment problem is $\mathcal{NP}$-hard.*

Of course we have $\mathcal{M}_2 \supset \mathcal{M}_1 \supset \mathcal{M}$. Even the class $\mathcal{M}$ is very broad; note that $\mathcal{M}$ contains each scoring matrix M for which there is $a_i \in \Sigma$ such that M penalizes mismatches of a_i with some $a_j \in \Sigma$ relative to a_i–a_i and a_j–a_j matches, penalizes all spaces aligned with a_i more heavily than mismatches between a_i and a_j, and penalizes all spaces to some extent. Thus $\mathcal{M}$ appears to cover most scoring schemes used in biological applications. A notable exception are scoring schemes that use a fixed gap penalty or a fixed penalty for gaps that exceed a specified length. Our proof will implicitly show that the gap-0-1 multiple alignment problem for the latter scoring schemes is still $\mathcal{NP}$-hard, but the question remains open for gap-0 multiple alignment and multiple alignment.

Some soring schemes used in practice do not penalize insertion of spaces at the beginning and end of sequences. While such scoring schemes do not formally belong to the classes $\mathcal{M}_2$, $\mathcal{M}_1$ and $\mathcal{M}$, it will be clear from the proofs that the analogue of Theorem 1 remains valid for them.

---

equal length is computationally intractable, then so is the more general problem of multiple alignment of sequences of "roughly equal" length.

[2] In matrices of practical interest, $x = 0$. Our proofs work regardless of whether $x = 0$ or $x > 0$.

We will also consider the following scoring matrix M_0 for the alphabet $\Sigma_0 = \{A, T, C\}$:

	Δ	A	T	C
Δ	0	2	2	2
A	2	0	1	0
T	2	1	0	0
C	2	0	0	0

FIG. 2. The scoring matrix M_0.

This scoring matrix does belong to $\mathcal{M}$, but it does not satisfy the triangle inequality and thus is not metric.

Some $\mathcal{NP}$-hard optimization problems have so-called *polynomial time approximation schemes* (abbreviated PTAS), that is, for every $\varepsilon > 0$ there exists a polynomial-time algorithm A_ε that is guaranteed to find for each instance a solution that is within a factor of $1 + \varepsilon$ of the optimal solution for this instance.[3] It can be shown that if an optimization problem belongs to a class called *MAX-$\mathcal{SNP}$-hard* problems, then it does not have a PTAS (unless $\mathcal{P} = \mathcal{NP}$) (Arora *et al.* 1992).

Theorem 2. *For the three-letter alphabet Σ_0 and the scoring matrix M_0 defined above, each of the following problems is MAX-$\mathcal{SNP}$-hard:*
(a) *The multiple alignment problem.*
(b) *The gap-0 multiple alignment problem.*
(c) *The gap-0-1 multiple alignment problem.*

It is not known whether there exists a scoring matrix N that is a metric such that the multiple alignment problem, the gap-0 alignment problem, or the gap-0-1 multiple alignment problem for N is MAX-$\mathcal{SNP}$-hard (Jiang *et al.* 1999). This question is open even if one only requires that all diagonal entries are zero, whereas all off-diagonal entries are positive (Della Vedova 1999).

3. Proofs

We will prove Theorems 1 and 2 by reducing the SIMPLE MAX-CUT(B) problem to the respective multiple alignment problems. Recall that an instance of size k of the SIMPLE MAX-CUT(B) problem is a simple graph $G = \langle V, E\rangle$ such that $|V| = k$ and each vertex of G has degree at most B. The problem is to find a partition of the set of vertices V into disjoint sets V_0 and V_1 such that the number of edges that connect a vertex in V_0 with a vertex in V_1, i.e., the size of the *cut determined by* $\langle V_0, V_1\rangle$, is as large as possible. There exists a fixed positive integer B such that the SIMPLE MAX-CUT(B) problem is $\mathcal{NP}$-hard; in fact, $B = 3$ works (Garey and Johnson 1979).

Proof of Theorem 1. Clearly, if the gap-0 multiple alignment problem is $\mathcal{NP}$-hard for each generic scoring matrix M with $z > v_T$, then the gap-0 multiple alignment problem is $\mathcal{NP}$-hard for all matrices in $\mathcal{M}_1$. Analogous observations can be made for $\mathcal{M}_2$ and $\mathcal{M}$. This allows us to prove Theorem 1 by proving $\mathcal{NP}$-hardness of

[3] Many authors use a slightly more stringent definition of a PTAS that requires ε to be a parameter of a single algorithm. But MAX-$\mathcal{SNP}$-hardness implies the nonexistence even of the weak kind of PTAS defined here.

the multiple alignment problems mentioned in it for the respective generic scoring matrices M.

Let k be a positive integer, and let B be such that the SIMPLE MAX-CUT(B) problem is $\mathcal{NP}$-hard. Given a graph $G = \langle V, E\rangle$ with k vertices and degree at most B, we define a k^2-tuple $\bar{t}^G = \langle t_0, \ldots, t_{k^2-1}\rangle$ of sequences as follows: Enumerate $V = \{v_0, \ldots, v_{k-1}\}$, $E = \{e_0, \ldots, e_{\ell-1}\}$. Each sequence t_i will have length $k^{12}\ell$. Intuitively speaking, for $i < k$, the sequence t_i will encode the vertex v_i. Sequences t_i for $i \geq k$ will be dummy sequences consisting entirely of T's. The role of the latter is to ensure that undesirable alignments are heavily penalized. Edge $e_m = \{v_i, v_h\}$ will be encoded by characters $t_{h,j}, t_{i,j}$, where $j = k^7\ell n + k^7 m + r$, $n < k^5$, $r \in \{1, 2, 3\}$. More precisely, we define $t_{i,j}$, the j-th character in t_i, as follows. For $m < \ell$, $e_m = \{v_h, v_i\}$, $h < i$, $n < k^5$ we let:
$t_{h,k^7\ell n+k^7 m+2} = t_{i,k^7\ell n+k^7 m+1} = t_{i,k^7\ell n+k^7 m+3} = A$.
In all other cases, we let $t_{i,j} = T$.

Figure 3 illustrates this construction. We exhibit a situation where $e_m = \{v_h, v_i\}, e_{m'} = \{v_g, v_h\}$, $m < m'$, $n < n' < k^5$.

		$t_{g,k^7\ell n+k^7 m}$						$t_{g,k^7\ell n+k^7 m'}$						$t_{g,k^7\ell n'+k^7 m}$					
		↓						↓						↓					
t_g:	...	T	T	T	T	T	...	T	T	A	T	T	...	T	T	T	T	T	...
		|	|	|	|	|		|	|	|	|	|		|	|	|	|	|	
t_h:	...	T	T	A	T	T	...	T	A	T	A	T	...	T	T	A	T	T	...
		|	|	|	|	|		|	|	|	|	|		|	|	|	|	|	
t_i:	...	T	A	T	A	T	...	T	T	T	T	T	...	T	A	T	A	T	...
		|	|	|	|	|		|	|	|	|	|		|	|	|	|	|	
t_p:	...	T	T	T	T	T	...	T	T	T	T	T	...	T	T	T	T	T	...

FIG. 3. Coding a graph in the proof of Theorem 1.

Now consider a gap-0-1 alignment a of the sequences $\bar{t}^G$. Such an alignment naturally induces a partition of V into disjoint subsets V_0^a and V_1^a, where V_1^a consists of all vertices v_i such that a appends a space at the beginning of t_i (i.e., shifts t_i to the right) and V_0^a consists of all vertices v_i such that a appends a space at the end of t_i (i.e., t_i remains in place). Let c_a denote the number of edges in G that connect vertices in V_a^0 with vertices in V_1^a, i.e., c_a denotes the size of the cut induced by the partition $\langle V_0^a, V_1^a\rangle$. We will show that if k is sufficiently large (i.e., $k \geq k_0$ for some fixed k_0) and a is an optimal gap-0-1 alignment for a generic matrix M of the sequences $\bar{t}^G$, then c_a is maximal. To see that this suffices for the proof of Theorem 1(c), note that the partition $\langle V_0^a, V_1^a\rangle$ can be decoded from a by a polynomial-time algorithm and every partition of V can be represented as $\langle V_0^a, V_1^a\rangle$ for a suitable gap-0-1 alignment a. It follows that if there exists a polynomial-time algorithm A for gap-0-1 alignment with respect to M, then a polynomial-time algorithm for the SIMPLE MAX-CUT(B) problem can be obtained as follows: For

graphs with $k \geq k_0$ vertices, encode the graph as a multiple sequence alignment problem in the way described above, run algorithm A to find the optimal gap-0-1 alignment, and then decode the partition $\langle V_0^a, V_1^a \rangle$ from the alignment. For the finitely many graphs of degree $\leq B$ with fewer than k_0 vertices, construct a lookup table of optimal solutions of the SIMPLE MAX-CUT problem, and use it for the algorithm. Note that using the lookup table only adds a constant (although possibly a large one) to the execution time of the algorithm. Throughout the remainder of this paper, we will without further comments always assume that k is "sufficiently large."

So let M be a generic scoring matrix. Let us estimate the SP-score for the aligned sequences $\langle at_0, \ldots, at_{k-1} \rangle$. This score has two components: indel (plus possibly gap opening) penalties and scores for character matches/mismatches. Since indel and gap opening penalties occur only in the first and last columns, the total of those penalties will be of order $O(k^4)$, which for sufficiently large k will be negligible. Recall that u, the penalty for A-T mismatches, was assumed to be greater than $\max\{0, v_A, v_T\}$. The total number of character mismatches in the unaligned sequences is $3k^5\ell(k^2-1)$. The idea of the proof is to find a gap-0-1 alignment a that maximally reduces this number by creating as many A-A matches as possible. A gap-0-1 alignment can create an A-A match only if the two A's are in adjacent columns, and each such newly created match will eliminate precisely two A-T mismatches. Note that whenever $e = \{v_h, v_i\} \in E$ and v_h, v_i end up in different parts of the partition $\langle V_0^a, V_1^a \rangle$ (i.e., the edge e is cut by the partition), then a total of k^5 A-A matches between sequences t_h and t_i are created, that is, $2k^5$ A-T mismatches between these sequences are eliminated. No other A-T mismatches can be eliminated by a gap-0-1 alignment, nor can a gap-0-1 alignment introduce additional A-T mismatches. It follows that the total SP-score for the aligned sequences is equal to

$$k^{12}\ell v_T k^2(k^2-1)/2 + 3k^5\ell(u - v_T)(k^2-1) - c_a k^5(2u - v_A - v_T) + O(k^4),$$

and thus for sufficiently large k, the optimal gap-0-1 alignment of $\bar{t}^G$ yields a partition of V that maximizes c_a.

For the proof of Theorem 1(b), let M be a generic scoring matrix with $z > \max\{0, v_T\}$. We will refer to the vector $\langle at_{0,j}, \ldots, at_{k^2-1,j} \rangle$ of j-th characters of the aligned sequences as the *j-th column* of the alignment. Note that we can compute the SP-score (excluding gap opening penalties) of an alignment a as $\sum_i \sum_j 0.5 sc_a(t_{i,j})$, where i ranges of the sequences in the alignment, j ranges over the columns in the alignment, and $sc_a(t_{i,j})$ is the sum over all pairwise scores between $t_{i,j}$ and the other symbols in the same column. (In particular, if a_0 is the alignment without any space symbols, then $sc_{a_0}(t_{i,j}) = \sum_{i' \neq i} d_M(t_{i,j}, t_{i',j})$.)

Lemma 3. *If $z > \max\{0, v_T\}$ and a is an optimal gap-0 alignment or an optimal multiple alignment of the sequences $\bar{t}^G$, then at most $O(k^6)$ columns of a contain space symbols.*

Proof. Consider the alignment a_0 that does not contain any spaces whatsoever, and let a be an alignment with better score than a_0. Note that our assumption on z implies that the score for a_0 can be improved only by replacing some A-T mismatches by T-T matches, or, if $y < u$, by A-Δ matches. On the other hand, replacing any T-T match by a T-Δ match will worsen the score by $z - v_T$. Since

$\ell \leq Bk/2$, only $O(k^6)$ of the columns of the unaligned sequences contain any A's. Thus the maximum possible improvement in the score of a_0 that can be achieved by inserting spaces is of the order $O(k^8)$. For each column c of a, let us define the *net gain* contributed by this column as

$$ng(c) = \sum_{t_{i,j} \in c} sc_{a_0}(t_{i,j}) - sc_a(t_{i,j}).$$

Of course, a negative net gain is a net loss. Now suppose a column c of a contains at least one space symbol and $ng(c) \geq 0$. If $z > u$, then it is easy to see that this column must contain at least one occurrence of A. If $z \leq u$, then either c contains at least one occurrence of A, or c contains at most $\lfloor u/z \rfloor$ space symbols *and* at least $\lceil (k^2-1)(z-v_T)/(u-v_T) \rceil$ T's from columns of a_0 that contain an occurrence of A. Let us relax these requirements a little and say that column c of a is *benign* if either it contains an occurrence of A or c contains at most $2\lceil u/z \rceil + 1$ space symbols and at least $0.5\lceil (k^2-1)(z-v_T)/(u-v_T) \rceil$ T's from columns of a_0 that contain an occurrence of A. Then there are at most $O(k^6)$ benign columns in a, and each column that is *not* benign contributes a net loss of at least $0.5(k^2-1)\min\{z-v_T, (z-v_T)^2/(u-v_T)\}$. Since the total gain of order $O(k^8)$ must outweigh the combined net loss of all columns, we conclude that all but $O(k^6)$ columns of a are benign, and the lemma follows. □

The definition of the partition $\langle V_0^a, V_1^a \rangle$ for a gap-0-1 alignment a of $\bar{t}^G$ can be generalized to gap-0 alignments in a natural way. In the latter case, V_0^a will consist of all vertices v_i such that a appends an even number of spaces at the beginning of t_i, and V_1^a will consist of all vertices v_i such that a appends an odd number of spaces at the beginning of t_i. For each gap-0 alignment a one can define a gap-0-1 alignment a^* that appends a space at the beginning of t_i if and only if $i < k$ and $v_i \in V_1^a$. Then $V_0^a = V_0^{a^*}$ and $c_a = c_{a^*}$. Let a_0 denote the alignment that contains no spaces, and let us analyse how much the SP-score of a_0 can be reduced by an optimal gap-0 alignment a. The total penalty for A-T mismatches can be reduced by creating A-A matches or, if $y < u$, by shifting some offending A's to the side where they are aligned with spaces rather than T's. The A's come in groups of three that reside in consecutive columns of a_0 and are separated by spacers of length $k^7 - 3$. Lemma 3 implies that a can shift sequences only by distances that are much shorter than the spacers. It follows that a can create matches only between two A's that sit in adjacent columns of a_0, and a cannot reduce penalties by shifting more than the three leftmost A's "to the side." But for each match between A's from neighboring columns of a_0 that is created by a, such a match is also created by a^*. Thus, the SP-score for the optimal gap-0 alignment a will again be equal to

$$k^{12}\ell v_T k^2(k^2-1)/2 + 3k^5\ell(u-v_T)(k^2-1) - c_a k^5(2u - v_A - v_T) + O(k^4),$$

and a induces a partition of V that maximizes c_a, which implies Theorem 1(b).

Finally, let M be a generic scoring matrix with $y > u$ and $z > v_T$, and let a be a multiple alignment that minimizes $SP^M(at_0, \ldots, at_{k^2-1})$. Let us think of the sequences $\bar{t}^G$ as forming k^5 consecutive blocks, where block number n consists of all columns of a_0 numbered $k^7\ell n$ through $k^7\ell(n+1)$. For $0 < n < k^5 - 1$, let us refer to columns numbered $k^7\ell n - \lfloor k^7/2 \rfloor$ through $k^7\ell(n+1) - \lfloor k^7/2 \rfloor - 1$ of a as *a-block number n*. Furthermore, *a-block number* 0 will consist of all positions to the left of a-block number 1, and *a-block number* $k^5 - 1$ will consist of all positions to

the right of a-block number k^5-2. Lemma 3 implies that for all n, the A's from block number n of the unaligned sequences must end up in a-block number n of the aligned sequences $\langle at_0,\ldots,at_{k^2-1}\rangle$.

Now let us consider a-block number n, which will be denoted by B_n, and let us estimate the combined net gain or net loss over all columns of B_n. There are two possibilities:

Case 1: B_n does not contain a space symbol.

In this case, we let $V_0^{a,n}$ be the set of all v_i such that a inserts an even number of space symbols into t_i to the left of B_n, and let $V_0^{a,n}$ be the set of all v_i such that a inserts an odd number of space symbols into t_i to the left of B_n. Let $c_{a,n}$ be the size of the cut determined by the partition $\langle V_0^{a,n},V_1^{a,n}\rangle$ of V. An argument as in the proof of part (b) shows that the combined net gain of all columns of a on B_n will be at most $2c_{a,n}(2u-v_A-v_T)$.

Case 2: B_n does contain a space symbol.

First note that insertion of space symbols might increase the number of A-A mismatches over what can be achieved by a gap-0 alignment, since the number of such matches will no longer be bounded by the size of any cut. However, Lemma 3 still implies that these matches have to be between A's from adjacent columns. Thus the number of A-A matches is bounded by ℓ; in other words, the combined net gain $sc_{a_0}(t_{i,j})-sc_a(t_{i,j})$ over all symbols $t_{i,j}$ in B_n is bounded by $2\ell(u-v_A-v_T)$, which is of order $O(k)$, since $\ell\le Bk/2$. Now let $\varepsilon=\min\{y-u,z-v_T\}$. Then any column that contains a space symbol contributes a net loss of at least $\varepsilon(k^2-1)-2u+v_A-v_T$, and it follows that the SP-score for a on B_n is worse than the SP-score for a_0 on B_n.

Now let us estimate the total SP-score for the alignment a. Let U be the set of all $n<k^5$ such that a-block number n does not contain spaces. Then

$$SP_a^M(\bar{t}^G)\ge \ell v_T(k^{16}-k^{14})/2+3k^5\ell(u-v_T)(k^2-1)-\sum_{n\in U}c_{a,n}k^5(2u-v_A-v_T)+O(k^4).$$

Let b be an optimal gap-0-1 multiple alignment of the sequences $\bar{t}^G$. Since the optimal multiple alignment a cannot have a score that is worse than that of an optimal gap-0-1 multiple alignment, we must have

$$SP_a^M(\bar{t}^G)\le v_T(k^{16}-k^{14})/2+3k^5\ell(u-v_T)(k^2-1)-2c_bk^5(2u-v_A-v_T)+O(k^4).$$

It follows that $c_{a,n}=c_b$ for most n, and thus for most n the partition $\langle V_0^{a,n},V_1^{a,n}\rangle$ maximizes the size of the cut in G. Since the largest of the numbers $c_{a,n}$ and the corresponding partition $\langle V_0^{a,n},V_1^{a,n}\rangle$ can easily be extracted from a by a polynomial-time algorithm, part (a) of Theorem 1 follows. □

Proof of Theorem 2. Our argument does not require a formal definition of the class MAX-$\mathcal{SNP}$. It suffices to know that there is a positive integer B such that the SIMPLE MAX-CUT(B) is MAX-$\mathcal{SNP}$-complete (Papadimitriou and Yannakakis 1991). We will show MAX-$\mathcal{SNP}$-hardness of our multiple alignment problems by showing that there are L-reductions of the SIMPLE MAX-CUT(B) problem to scaled versions of each of the following problems: gap-0-1 multiple alignment for M_0, gap-0 multiple alignment for M_0, and multiple alignment for M_0. This establishes MAX-$\mathcal{SNP}$-hardness in the sense of Arora and Lund (1997), who call an optimization problem Π' *MAX-$\mathcal{SNP}$-hard* if there exist a MAX-$\mathcal{SNP}$-complete problem Π and a gap-preserving reduction of Π to Π'. This a definition explicitly allows scaling of objective functions (see Arora and Lund (1997), page 411).

Let us recall the notion of an *L-reduction.* If Π and Π' are two optimization (maximization or minimization) problems, then Π *L-reduces* to Π' if there are two polynomial-time algorithms f, g and constants $\alpha, \beta > 0$ such that for each instance I of Π:
(a) Algorithm f produces an instance $I' = f(I)$ of Π', such that the optima of I and I', $OPT(I)$ and $OPT(I')$, respectively, satisfy $OPT(I') \leq \alpha OPT(I)$.
(b) Given any solution of I' with cost c', algorithm g produces a solution of I with cost c such that $|c - OPT(I)| \leq \beta |c' - OPT(I')|$.

Let us define a minimization problem Π' as follows: An instance of Π' is a simple graph $G = \langle V, E \rangle$ with degree at most B. For every partition $P = \langle V_0, V_1 \rangle$ of V, let c_P be the size of the cut determined by P. The objective of Π' is to find a partition P of V that minimizes the number $d_P = 3|E| - 2c_P$.

Here is the L-reduction of Π' to scaled versions of the multiple alignment problems: Given a graph $G = \langle V, E \rangle$ with k vertices and degree at most B, we define a k^2-tuple $\bar{t}^G = \langle t_0, \ldots, t_{k^2-1} \rangle$ of sequences as follows: Enumerate $V = \{v_0, \ldots, v_{k-1}\}$, $E = \{e_0, \ldots, e_{\ell-1}\}$. Each sequence t_i will have length $k^{12}\ell$. We define $t_{i,j}$, the j-th character in t_i, as follows. For $m < \ell$, $e_m = \{v_h, v_i\}$, $h < i$, $n < k^5$ we let:
$t_{h,k^7\ell n + k^7 m + 2} = t_{i,k^7\ell n + k^7 m + 1} = t_{i,k^7\ell n + k^7 m + 3} = A$.
$t_{h,k^7\ell n + k^7 m + 1} = t_{h,k^7\ell n + k^7 m + 3} = t_{i,k^7\ell n + k^7 m + 2} = T$.
In all other cases, we let $t_{i,j} = C$.

Figure 4 illustrates this construction. Again, we exhibit a situation where $e_m = \{v_h, v_i\}, e_{m'} = \{v_g, v_h\}$, $m < m'$, $n < n' < k^5$.

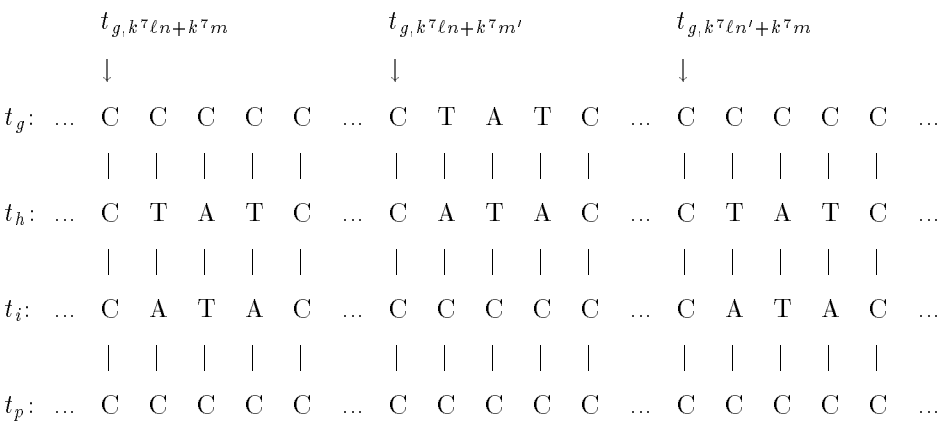

FIG. 4. Coding a graph in the proof of Theorem 3.

An argument very similar to the reasoning in the proof of Theorem 1 shows that if a is the optimal gap-0-1 multiple alignment, gap-0 multiple alignment, or multiple alignment for M_0, then

$$(*) \qquad SP_a^{M_0}(\bar{t}^G) = (3\ell - 2c_a)k^5 + O(k^4),$$

where c_a is the size of the minimal cut in G.

Now it is immediately clear that Π' L-reduces to each of the three alignment problems, if the SP-score is scaled by a factor of $k^{-5/2}$ for every multiple alignment problem that involves k sequences.

Since L-reductions compose, it now suffices to show that the SIMPLE MAX-CUT(B) problem L-reduces to Π'. Let $G = \langle V, E \rangle$ be a simple graph of degree at most B. The functions f and g in the definition of an L-reduction will simply be identity. Note that for any partition P of V that maximizes the size of the corresponding cut, each vertex of degree ≥ 1 contributes at least one adjacent edge to the cut induced by P: If not, the size of the cut could be increased by moving the offending vertex to the other side of the partition. It follows that if the degrees in G are bounded by B, then $c_P \geq |E|/B$. Since $d_P = 3|E| - 2c_P \leq 3|E|$, we can set $\alpha = 3B$. Since any increase of c_P by 1 corresponds to a decrease of d_P by 2, we can set $\beta = 2$, and the conditions of an L-reduction will be satisfied.

□

4. Acknowledgements

I would like to thank Liming Cai for bringing the problem to my attention, and David Juedes, Gianluca Della Vedova and Tao Jiang for valuable comments on earlier versions of this paper and a prequel (Just 1999) to it. I also thank the referee of the first version of this paper for pointing out a mistake in the argument.

5. References

Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M. 1992. Proof verification and intractability of approximation problems, 13-22. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science.*

Arora, S., and Lund, C. 1997. Hardness of Approximations, 399-446. *In* Hochbaum, D.S., ed., *Approximation Algorithms for NP-hard Problems,* PWS.

Benner, S.A., Cohen, M.A., and Gonnet, G.H. 1993. Empirical and Structural Models for Insertions and Deletions in the Divergent Evolution of Proteins. *J. Mol. Biol.* 229, 1065-1082.

Bonizzoni, P., and Della Vedova G. 2000. The complexity of multiple alignment with SP-score that is a metric. To appear in *Theoretical Computer Science.*

Carillo, H. and Lipman, D. 1988. The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.* 48(5), 1073-1082.

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. 1978. A model of evolutionary change in proteins. *In* M. O. Dayhoff, ed., *Atlas of Protein Sequence and Structure,* 345-352. Nat. Biomed. Res. Found., 5, supp. 3.

Della Vedova, G. 1999. *Personal Communication.*

Fasman, K.H., and S. L. Salzberg, S.L. 1998. An introduction to biological sequence analysis, 21-42. *In* Salzberg, S.L., Searls, D.B., and Kasif, S., eds., *Computational Methods in Molecular Biology,* Elsevier.

Fitch, W. M. 1993. Letter to the Editor: Commentary on the letter by Ward C. Wheeler. *Molecular Biology and Evolution* 10(3), 713-714.

Garey, M.R., and Johnson, D.S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman.

Henikoff, S., and Henikoff, J. 1992. Amino acid substitution matrices from protein blocks. *Proc. Nat. Acad. of Sci., USA,* 89, 10915-10919.

Jiang, T. 1999. *Personal Communication.*

Jiang, T., Kearney, P., and Li, M. 1999. Some Open Problems in Computational Molecular Biology. SIGACT News 30(3), 43-49.
Just, W. 1999. On the computational complexity of gap-0 multiple alignment. *Preprint.*
Papadimitriou, C., and Yannakakis, M. 1991. Optimization, approximation and complexity classes. *J. of Computer and System Sciences* 43, 425-440.
Pascarella, S., and Argos, P. 1992. Analysis of Insertions/Deletions in Protein Structures. *J. Mol. Biol.* 224, 461-471.
Wang, L., and T. Jiang, T. 1994. On the complexity of multiple sequence alignment. *Journal of Computational Biology* 1(4), 337-348.
Wheeler, W. C. 1993. Letter to the Editor: The Triangle Inequality and Character Analysis. *Molecular Biology and Evolution* 10(3), 707-712.

	Δ	A	T
Δ	x	y	z
A	y	v_A	u
T	z	u	v_T

FIG. 1. A generic scoring matrix.

	Δ	A	T	C
Δ	0	2	2	2
A	2	0	1	0
T	2	1	0	0
C	2	0	0	0

FIG. 2. The scoring matrix M_0.

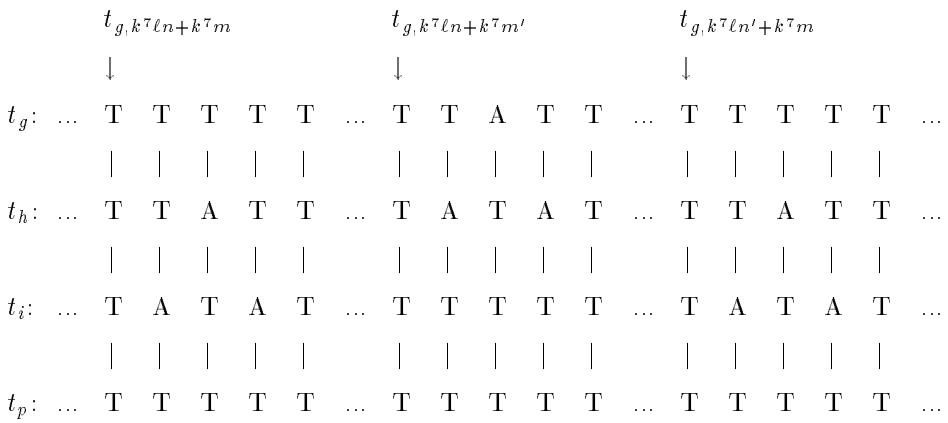

FIG. 3. Coding a graph in the proof of Theorem 1.

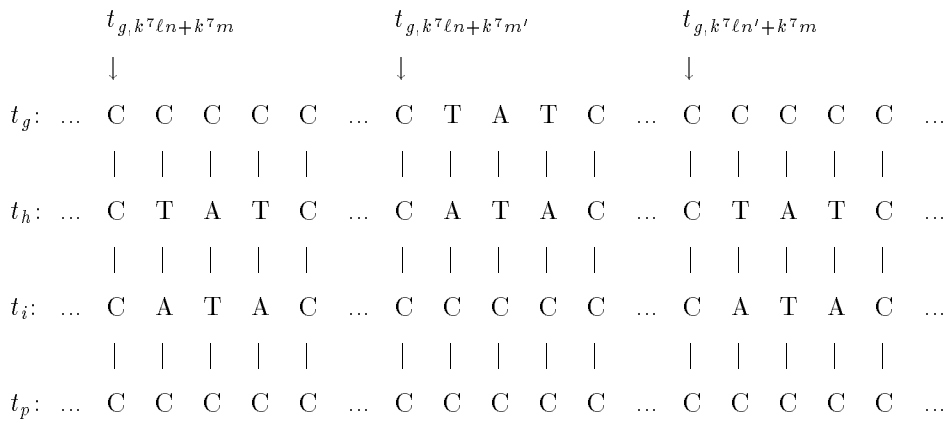

FIG. 4. Coding a graph in the proof of Theorem 3.