Multiple Sequence Alignment as a Facility Location Problem[1]

Winfried Just and Gianluca Della Vedova

Department of Mathematics, Ohio University, Athens, Ohio 45701, U.S.A.
Dipartimento di Informatica, Sistemistica e Comunicazione,
Universitá degli Studi di Milano - Bicocca, 20126 Milano, Italy

e-mail: `just@math.ohiou.edu, dellavedova@disco.unimib.it`

Abstract. A connection is made between certain multiple sequence alignment problems and facility location problems, and the existence of a PTAS (polynomial time approximation scheme) for these problems is shown. Moreover, it is shown that multiple sequence alignment with SP-score and fixed gap penalties is MAX SNP-hard.

Key words: multiple sequence alignment, SP-score, Space-L Alignment problem, Switchboard Location problem, PTAS, smooth polynomial programming

1 Introduction

Recent advances in the availability of biological data (i.e. DNA, RNA or protein) has led to tremendous improvements in Molecular Biology. This huge amount of data has also given a tremendous boost to a new field of Computer Science called Bioinformatics. Pattern matching is a basic tool in Molecular Biology, as sequence similarity usually implies homology and functional similarity of the proteins or genes encoded by such sequences. Another crucial application of sequence comparison are searches of biological databases. All known biological sequences are stored in huge databases (e.g. EMBL, Swiss-Prot), and all recent papers in Molecular Biology that report the discovery of a new sequence include a detailed comparison of the novel sequences with those stored in the publicly available databases.

These facts reveal the importance of developing efficient algorithms for aligning a set of sequences. It is standard practice to represent biological sequences as sequences over a fixed alphabet (4 symbols for DNA and RNA sequences, 20 symbols for proteins). An alignment of a set $\mathcal{S}$ of sequences is basically a matrix where the rows correspond to the sequences in the set, possibly with some spaces inserted, and the cost of an alignment is the sum of the costs of all columns. The goal is to compute the alignment of $\mathcal{S}$ of minimum cost (or, in an equivalent formulation preferred by many biologists, maximum score). This general definition allows different specifications of the problem, according to the definition of cost of a column in the alignment we choose. In practice at least two definitions make sense, the first (called Tree Alignment) requires a tree whose node set is exactly $\mathcal{S}$ and where the cost of a column

[1]The second author was supported by MURST grant "Bioinformatica e ricerca genomica."

is the sum of the costs of the pairs of symbols of the two sequences that are adjacent in the tree. A particular case of this problem is called Star Alignment, which is the restriction to trees with exactly one internal node. The other definition (called SP-Alignment) will be the one studied in this paper and defines the cost of a column as the sum of all pairs of symbols in the column. Equivalently, the SP-score can be defined as the sum over the pairwise alignment scores of all induced alignments of pairs of the sequences. The pairwise alignment scores are defined as follows. Let Σ be a fixed alphabet and let $\Delta \notin \Sigma$ denote the space symbol, then a *scoring scheme* is a symmetric *scoring function* $d_M : (\Sigma \cup \Delta) \times (\Sigma \cup \Delta) \mapsto \mathbb{N}$ together with specifications on how to handle gaps. A scoring function d_M can be conveniently represented by a *scoring matrix* M. The cost of a pair of symbols s_1, s_2 under the scoring matrix M is $d_M(s_1, s_2)$. A *gap* is a string of the form Δ^i. Most scoring schemes used in practice are *affine*, i.e., they specify a fixed *gap opening penalty* g (possibly 0) that is added to the score calculated according to d_M for each newly created gap in the alignment. In this context, the numbers $d_M(s, \Delta)$ for $s \in \Sigma$ are called *gap extension penalties.* Note that if all gap extension penalties are zero, then we have a scoring scheme with *fixed gap penalties.* If $d_M(s, \Delta) > 0$ for all $s \in \Sigma$, then we will say that the scoring scheme specifies *strictly positive gap extension penalties.*

Both Tree Alignment and SP-Alignment problems have been proved to be NP-hard by Wang and Jiang [WJ94]. Hence reasearch has focused on heuristic algorithms or approximation algorithms for such problems and on finding restrictions that are efficiently solvable. A restriction which has a natural interpretation is the one where the scoring function is a metric. For this restriction some approximation algorithms with guaranteed $2-o(1)$ error ratio have been described [G93, P92, BLP97]. Moreover, optimal alignments in actual biological sequences tend to have relatively few gaps [W93, F93, BCG93, PA92], but unfortunately even in such restricted cases the SP-Alignment problem is still NP-hard [BD00, J99]. In [J99], several such modifications of the SP-Alignment problem were studied. In the Gap-0 Alignment problem, spaces may be inserted at the beginning and at the end of sequences, but not between characters from Σ, and the Gap-0-1 Alignment problem is the restriction of Gap-0 Alignment where at most one space can be inserted in each sequence. It turns out that SP-Alignment, Gap-0 Alignment and Gap-0-1 Alignment problems are all NP-hard for practically every affine scoring scheme with strictly positive gap extension penalties used by molecular biologists [J99]. This leaves open the case of other ways of calculating the gap penalties that are sometimes used in Molecular Biology. In particular, this leaves open the interesting case of fixed gap penalties, where all gaps are penalized equally, no matter where they occur and how long they are.

Moreover, it had been shown in [J99] that for some scoring matrix M the three problems mentioned above are MAX SNP-hard. The scoring matrix M used in the the latter result does not penalize all character mismatches, and thus is not metric. In [JKL99], Jiang *et al.* ask whether a particular restriction of the SP-Alignment problem (the case of metric scoring matrix) has a polynomial time approximation scheme (PTAS), that is, there exists a polynomial time approximation algorithm for any fixed constant guaranteed error ratio. In [LMW99] a related question has been answered positively by showing that the Star-c-Alignment Problem (where the number of gaps in the pairwise alignment between any given sequence and the median sequence is bounded by a constant c) with Hamming distance admits a PTAS. In our paper we show that a different restriction of the problem admits a PTAS. More precisely, we show that a PTAS exists if the total number of spaces that can be inserted into each sequence is bounded and the ratio of the costs between each pairwise alignment is in a fixed interval. Our results trivially hold also for Gap-0-1 Alignment.

Moreover, we will show that at least for some scoring scheme with fixed gap penalty, the Gap-0 Alignment and the SP-Alignment problems are MAX SNP-hard. Since the optimal alignment in the example that yields the latter result contains only one space in each sequence, the requirement of bounded cost ratio cannot be dropped from the construction of the PTAS we describe in the paper.

2 Preliminaries

Let Σ be a finite alphabet, let $\Delta \notin \Sigma$ be the space symbol, let $d_M : (\Sigma \cup \Delta) \times (\Sigma \cup \Delta) \to \mathbb{N}$ be a function called *scoring function,* and let g be a nonnegative integer constant called *gap opening penalty.* The symbol $\alpha(M)$ will denote the maximum value $d_M(a_1, a_2)$ between two different symbols $a_1, a_2 \in \Sigma \cup \{\Delta\}$. Given a sequence a over $\Sigma \cup \{\Delta\}$, the symbol $a[i]$ will denote the i-th character of a and $|a|$ will denote the *length* n of a sequence $a = a[1], \ldots, a[n]$. Then given two sequences $s_1 = s_1[1], \ldots, s_1[m]$, $s_2 = s_2[1], \ldots, s_2[m]$ of m symbols over $(\Sigma \cup \Delta)$, the cost of aligning s_1 and s_2 is $d_M(s_1, s_2) = g(G_1 + G_2) + \sum_{i=1}^{m} d_M(s_1[i], s_2[i])$, where G_j is the number of gaps (consecutive runs of space symbols) in s_j. Given a k-tuple $< t_1, \ldots, t_k >$ of sequences over the alphabet $\Sigma \cup \{\Delta\}$, a *multiple alignment* is a k-tuple $< at_1, \ldots, at_k >$ of equal-length sequences (where at_i stands for *aligned* t_i) over the alphabet $\Sigma \cup \{\Delta\}$ such that each at_i can be obtained from t_i by inserting some space symbols into the sequences without altering the order of symbols of t_i. Given two equal-length sequences at_1, at_2, their *pairwise alignment* is the pair of sequences bt_1, bt_2 that is obtained from at_1, at_2 by removing all columns containing only Δs. If L is a nonnegative integer, by $d^{opt}_{M,L}(t_1, t_2)$ we will denote the minimum cost among all pairwise alignments of $< t_1, t_2 >$ that insert at most L spaces into each of the sequences t_1, t_2. The SP-Alignment problem for a given scoring scheme (d_M, g) is to find the multiple alignment $< at_1, \ldots, at_k >$ that minimizes $SP(< at_1, \ldots at_k >) = \sum_{1 \le i < j \le k} d(bt_i, bt_j)$ among all possible multiple alignments of $< t_1, \ldots, t_k >$.

Here we will study a restriction of SP-Alignment that captures to some extent the pattern of space insertions observed in real biomolecular sequences and is different from the restrictions studied in [J99]. A *space-L alignment* $\mathcal{A}$ of a k-tuple $< t_1, \ldots, t_k >$ of sequences is an alignment $< at_1, \ldots, at_k >$ of $< t_1, \ldots, t_k >$ such that $|at_i| \le |t_i| + L$ for each sequence t_i. Note that space-L-alignments exist only if the length of the shortest of these sequences is at least $n - L$, where n is the length of the longest among the sequences $t_1, \ldots, t_k$. Please also note that there are no restrictions about where the space symbols can be inserted. The Space-L Multiple Alignment problem asks to find, for a k-tuple of sequences $< t_1, \ldots, t_k >$ and a scoring scheme (d_M, g), a space-L multiple alignment that minimizes the SP-score with respect to (d_M, g). Given an instance $I =<< t_1, \ldots, t_k >, (d_M, g) >$ of the Space-L Multiple Alignment problem we define the *variability* of I, denoted by $v(I)$, as

$$v(I) = \max\{\frac{n\alpha(M) + Lg}{d^{opt}_{M,L}(t_i, t_j)} : 1 \le i < j \le k\},$$

Please note that the value $v(I)$ of the instance I can be computed in polynomial time. The Space-L Multiple Alignment(σ) problem is the restriction of the Space-L Multiple Alignment problem to instances I with $v(I) \leq \sigma$.

A few comments are in order. The most common multiple alignment problem in Molecular Biology is the alignment of homologous protein sequences from different species. For a pair $< t_i, t_j >$ of such sequences, $< a(i,j)t_i, a(i,j)t_j >$ will be small only if the sequences are very similar, which usually happens only if the two species of origin have a relatively recent (in the timescale of evolution) common ancestor, and will be close to the average distance of random sequences if the species diverged a long time ago, or if the optimal alignment requires more than L spaces. For scoring matrices used in practice, the average distance of random sequences is usually a number of about the same order of magnitude as $n\alpha(M)$. The algorithms used in practice for multiple sequence alignment tend to perform well if all sequences are closely related to each other, while our first theorem covers one of the cases that are difficult in practice and quite common, namely the case where *none* of the sequences are closely related to each other.

3 The PTAS

The main results of this section is the following:

Theorem 1 *Let σ be a constant. Then the* Space-L Multiple Alignment*(σ) problem has a polynomial time approximation scheme.*

Note that in the above theorem, the scoring scheme (d_M, g) is considered part of the input, thus the theorem works for all affine scoring schemes, no matter whether the scoring function is a metric and the gap penalties are fixed or variable. This does not contradict the results about MAX SNP-hardness from [J99] though, since the variability of the instances used to obtain the latter results was not bounded.

Theorem 1 will be proved by reformulating it as a kind of facility location problem. To see the connection, suppose a communication network is to be set up in a country that consists of k regions. In each region, there should be one switchboard of the network, and each switchboard is to be connected by expensive, high quality cable to every other switchboard. If in each region there are several possible locations for the switchboard that are equally good for the operation of the network within this region, then the locations of switchboards should be chosen in such a way as to minimize overall cost of cable between them. The question of choosing optimal locations for the switchboards can then be formalized as follows. The Switchboard Location problem has as instance some disjoint sets $R_1, \ldots, R_k$ called *regions*, as well as a distance function d between all pairs of points x_i, x_j in $R_1 \cup \cdots \cup R_k$. The distance function gives strictly positive values whenever the two points are distinct. A feasible solution is a k-tuple $< x_1 \ldots, x_k >$ of points such that $x_i \in R_i$ for $1 \leq i \leq k$. The problem asks for a feasible solution that minimizes $\sum_{1 \leq i < j \leq k} d(x_i, x_j)$.

While facility location problems with objective functions similar to those of Switchboard Location have been studied for regions of the real line (see e.g. [AH97], [T94]), we are not aware of any published results concerning the general formulation of Switchboard Location given above.

We will discuss later how instances of Space-L Alignment(σ) can be mapped to suitable instances of Switchboard Location in order to have a $(1+\epsilon)$ approximation algorithm. But first we have to introduce a restriction of Switchboard Location similar to the one introduced for Space-L Alignment. Let $I = \{R_1, \ldots, R_k, d\}$ be an instance of the Switchboard Location problem. We define the *spread* $s(I)$ of I as

$$s(I) = \frac{\max\{d(x_i, x_j) : 1 \le i < j \le k,\ x_i \in R_i, x_j \in R_j\}}{\min\{d(x_i, x_j) : 1 \le i < j \le k,\ x_i \in R_i, x_j \in R_j\}}.$$

It is immediate from the definition that $s(i) \ge 1$. For any pair of constants P, σ, the Switchboard Location$_P(\sigma)$ problem is the Switchboard Location problem restricted to instances of spread at most σ and where each region contains at most P points.

Theorem 2 *Let P, σ be two constants. Then the* Switchboard Location$_P(\sigma)$ *problem admits a PTAS.*

Proof. The PTAS for Switchboard Location is based on the smooth polynomial programming technique of Arora *et. al* [AKK99]. We will briefly recall the relevant material from those papers. A *c-smooth polynomial integer program* (or PIP) is a problem of the form

$$\begin{array}{ll} \text{minimize} & p_0(x_1, \ldots, x_n) \\ \text{subject to} & l_j \le p_i(x_1, \ldots, x_n) \le u_j \\ & x_i \in \{0,1\} \text{ for } i = \{1, \ldots, n\} \end{array} \tag{1}$$

where each p_j is an n-variate polynomial of maximum degree d, and each coefficient of each degree ℓ monomial (term) has an absolute value of at most $c \cdot n^{d-\ell}$.

The fundamental result that we will use, Theorem 1.10 of [AKK99], asserts that, for each $\delta > 0$, there exists an approximation algorithm running in time $O(n^{\frac{1}{\delta^2}})$ that computes a 0/1 assignment $< y_1, \ldots, y_n >$ to the variables x_i of a c-smooth PIP such that, for n-variate degree-d polynomials, the value of $p_0(y_1, \ldots, y_n)$ is within an *additive* error is at most δn^d of minimum for 0/1 solutions that satisfy all constraints $p_1, \ldots, p_m$, and such that $< y_1, \ldots, y_n >$ satisfies each linear constraint within an additive error of $O(\delta\sqrt{n \log n})$.

Now let σ be a fixed constant, and suppose we have an instance I of the Switchboard Location$_P(\sigma)$ problem, where $\{R_i : 1 \le i \le k\}$ are the regions of I, and $R_i = \{x_{i,j} : 1 \le j \le P\}$. (Since one can always add dummy points to the regions, we do not lose generality by assuming the regions to be *exactly* of cardinality P.) Let D be the value $\min\{d(x_{i,j}, x_{h,\ell}) : 1 \le i < h \le k,\ 1 \le j, \ell \le P\}$. Now we can formulate the Switchboard Location problem as a PIP:

$$\begin{array}{lll} \text{minimize} & \sum_{1 \le h < i \le k,\ 1 \le j, \ell \le k} \frac{d(x_{i,j}, x_{h,\ell})}{D} y_{i,j} y_{h,\ell} & \\ \text{subject to} & \sum_j k y_{i,j} = k & i = 1, \ldots, k \\ & y_{i,j} \in \{0,1\} & i = 1, \ldots, k;\ j = 1, \ldots, P \end{array} \tag{2}$$

Please note that the total number of variables is at most kP. Since $s(I) \le \sigma$, all coefficients of the objective functions are between 1 and σ. Thus the PIP is σ-smooth.

Now suppose we want to find a solution to the Switchboard Location problem that is within a factor of $(1+\epsilon)$ of minimum. Setting $\delta = \frac{\epsilon}{2P^2}$, and running the algorithm of [AKK99] on the PIP defined above, we find a 0/1 solution that satisfies all constraints within an additive error of $O(\delta\sqrt{kP\log kP})$. Since for 0/1 solutions the left hand sides of our side constraints are multiples of k; for sufficiently large k we can assume that these side constraints are satisfied *exactly*. But then for each region R_i, exactly one of the numbers $y_{i,j}$ is equal to 1. Thus the corresponding $x_{i,j}$'s form a feasible solution of instance I of the Switchboard Location problem, and the sum of the distances is within an additive error of $D\epsilon\binom{k}{2}$. By the choice of D, the minimum value for the sum of all distances in any feasible solution of instance I of the Switchboard Location problem cannot be less than $D\binom{k}{2}$, and thus we have found, in polynomial time, an approximation within a factor of $(1+\epsilon)$. □

Now let us show how Theorem 2 implies Theorem 1. Suppose we are given an instance $I = << t_1, \ldots, t_k >, (d_M, g) >$ of the Space-L Alignment(σ) problem, and let $\epsilon > 0$. We want to find a space-L multiple alignment of these sequences that scores within $(1+\epsilon)$ of optimum. Let $N = \lceil \frac{4L\sigma}{\epsilon} \rceil$ and note that N is a constant. Let n be the length of the longest among the sequences $t_1, \ldots, t_k$, and let $K = \lceil 2N + \frac{gN}{\alpha(M)} \rceil$.

First assume that $n \leq K$. In this case we let R_i be the set of all sequences $x_{i,j}$ that are obtainable by inserting L spaces into t_i (at the beginning, end, or between symbols). This set contains at most $\binom{K+L}{L}$ elements. Note that $\binom{K+L}{L}$ is a constant that does not depend on the number of sequences k. Thus the family $\{R_i : 1 \leq i \leq k\}$ together with the distances $d(x_{i,j}, x_{i',j'})$ defined by the scoring scheme is an instance of the Switchboard Location problem where the cardinality of all regions is bounded by the constant $\binom{K+L}{L}$. Feasible solutions of the Switchboard Location problem are exactly all space-L alignments of our sequences, and the objective function of the Switchboard Location problem is exactly the SP-score of the alignment. Since the variability of the Space-L Alignment problem is bounded by σ, the spread of the corresponding Switchboard Location problem that we just constructed is also bounded by σ. Thus the PTAS for Switchboard Location$_{\binom{K+L}{L}}(\sigma)$ finds a solution within $(1+\epsilon)$ of optimum.

Now assume that $n > K$. In this case we partition each sequence t_i into consecutive chunks $< s_{i,h} : 1 \leq h \leq N >$, where the length of each chunk differs from $\frac{n}{N}$ by no more than 1. With each function $f : \{1, \ldots, N+1\} \to \mathbb{N}$ such that $\sum_{1 \leq i \leq N+1} f(i) \leq L$ we associate a sequence $t_{i,f}$ by inserting $f(h)$ space symbols to the left of each chunk $s_{i,h}$. In other words,

$$t_{i,f} = \Delta^{f(1)} s_{i,1} \Delta^{f(2)} s_{i,2} \ldots \Delta^{f(N)} s_{i,N} \Delta^{f(N+1)}$$

Now we let R_i be the set of all $t_{i,f}$ for functions $f : \{1, \ldots, N+1\} \to \mathbb{N}$ that satisfy $\sum_{1 \leq i \leq N+1} f(i) \leq L$. We run the approximation algorithm for Switchboard Location$_{N+1}(\sigma)$ that finds a solution within $(1+\frac{\epsilon}{3})$ on the instance given by the $k+1$-tuple $< R_1, \ldots, R_k, (d_M, g) >$.

The algorithm returns a space-L multiple alignment $< t_{1,f_1}, \ldots, t_{k,f_k} >$ of the sequences $< t_1, \ldots, t_k >$. It remains to show that the alignment $< t_{1,f_1}, \ldots, t_{k,f_k} >$ scores within $(1+\epsilon)$ of optimum. Let $< at_1, \ldots, at_k >$ denote a space-L multiple alignment with optimal SP-score. For each i, let $g_i : \{1, \ldots, N+1\} \to \mathbb{N}$ be the function such that for each $1 \leq i \leq k$ and $1 \leq h \leq N$, g_i is equal to the number of spaces in at_i inserted immediately to the left of the chunk $s_{i,h}$ or between characters of $s_{i,h}$. Instead of t_{i,g_i} we will write bt_i. Since $bt_i \in R_i$ for each i, we have

$$SP(< t_{1,f_1}, \ldots, t_{k,f_k} >) \leq (1 + \frac{\epsilon}{3}) SP(< bt_1, \ldots, bt_k >).$$

Since $1 + \epsilon > (1 + \epsilon/2)(1 + \epsilon/3)$ whenever $\epsilon < 1$, it now suffices to show that

$$SP(< bt_1, \ldots, bt_k >) \leq (1 + \frac{\epsilon}{2}) SP(< at_1, \ldots, at_k >).$$

Let us split the sequences at_i, bt_i into $N + 1$ chunks $at_{i,h}$, $bt_{i,h}$ for $1 \leq h \leq N + 1$ where $bt_{i,h} = \Delta^{g_i(h)} s_{i,h}$, $s_{i,N+1}$ is the empty string, and $|bt_{i,h}| = |at_{i,h}|$, so that $at_i = at_{i,1} at_{i,2} \cdots at_{i,N+1}$ and $bt_i = bt_{i,1} bt_{i,2} \cdots bt_{i,N+1}$. From the definition of g_i, whenever $g_i(h) = g_j(h) = 0$, the pairwise alignment $< at_{i,h}, at_{j,h} >$ is the same as $< bt_{i,h}, bt_{j,h} >$. Since at most L spaces are inserted into each sequence t_i, and since the maximum penalty on each chunk (excluding the newly inserted spaces) is equal to the length of the chunk (i.e. at most $\frac{n}{N} + 1$) multiplied by $\alpha(M)$, and there are globally only L extra spaces, we get the inequality

$$d_M(bt_i, bt_j) \leq d_M(at_i, at_j) + \alpha(M) L (2 + \frac{n}{N}) + Lg.$$

Since $n > 2N + \frac{gN}{\alpha(M)}$ and thus $\alpha(M)\frac{Ln}{N} \geq 2L\alpha(M) + Lg$, we get

$$d_M(bt_i, bt_j) \leq d_M(at_i, at_j) + \alpha(M) n \frac{2L}{N}.$$

By the choice of N, the latter yields

$$d_M(bt_i, bt_j) \leq d_M(at_i, at_j) + \frac{\alpha(M) n \epsilon}{2\sigma}.$$

Since the variability of our instance was assumed to be at most σ, the inequality $d_M(at_i, at_j) \geq \frac{\alpha(M)n}{\sigma}$ holds, and we get

$$d_M(bt_i, bt_j) \leq d_M(at_i, at_j)(1 + \frac{\epsilon}{2}),$$

as required.

4 MAX SNP-hardness

The following theorem shows that the assumption of bounded variability cannot be simply dropped in Theorem 1.

Theorem 3 *There exists a scoring scheme (d_M, g) with fixed gap penalties such that:*
(a) *For the scoring scheme (d_M, g) and for every $L > 0$ the* Space-L Multiple Alignment *problem is MAX SNP-hard.*
(b) *For the scoring scheme (d_M, g) the* Gap-0 Alignment *problem is MAX SNP-hard.*
(c) *For the scoring scheme (d_M, g), the* SP-Alignment *problem is MAX SNP-hard.*

Here is the scoring scheme mentioned in the above theorem. The alphabet will be $\Sigma = \{A, C, T\}$, the gap opening penalty will be $g = 2$, and the scoring matrix M will be:

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

Proof. We will prove Theorem 3 by reducing the Max Cut problem on cubic graphs (denoted by 3-Max Cut) to the respective multiple alignment problems. Recall that an instance of size k of the 3-Max Cut problem is a simple graph $G =< V, E >$ such that $|V| = k$ and each vertex of G has degree exactly 3. 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* $< V_0, V_1 >$, is as large as possible. It is well known that the 3-Max Cut problem is MAX SNP-hard [AK97]. For our purposes, it is most important to note that the latter implies that there exists a real $\epsilon > 0$ such that no polynomial-time approximation algorithm can find a cut such that the number of edges that are NOT cut is within an additive constant of ϵk of minimum.

Given a cubic graph $G =< V, E >$ with k vertices, we define a $2k$-tuple $\bar{t}^G =< t_1, \ldots, t_{2k} >$ of sequences as follows: Enumerate $V = \{v_1, \ldots, v_k\}$, $E = \{e_0, \ldots, e_{\ell-1}\}$. Each sequence t_i will have length $2(k^4 + \lceil \alpha k \rceil \ell)$, where α is a constant that will be defined below. Intuitively speaking, for $1 \le i \le k$, the sequence t_i will encode the vertex v_i. Edge $e_m = \{v_i, v_h\}$ will be encoded by characters $t_h[j], t_i[j]$, where $j = 2(\lceil \alpha k \rceil m+1), \ldots, 2\lceil \alpha k \rceil (m+1)$. More precisely, we define $t_i[j]$, the j-th character in t_i, as follows. For $1 \le m \le \ell$, $e_m = \{v_h, v_i\}$, $h < i$, $\lceil \alpha k \rceil m < j < \lceil \alpha k \rceil (m+1)$ we put: $t_h[2j] = A$, $t_i[2j] = T$, and $t_p[2j] = C$ for $p \notin \{i, h\}$.

The sequence t_{k+i} will act as a "mirror image" of t_i. The purpose of mirror images is to neutralize the effects of unbalanced cuts on the scores of aligments. For $1 \le i \le k$ and $\lceil \alpha k \rceil \ell + ik^3 < j < \lceil \alpha k \rceil \ell + (i+1)k^3$ we put: $t_i[2j] = A$, $t_{k+i}[2j] = T$, $t_p[2j] = C$ for $p \notin \{i, k+i\}$.

For all p, j, we let $t_p[2j-1] = C$. Let us illustrate this construction with a picture. We exhibit a situation where $e_m = \{v_h, v_i\}$.

	$t[2(\lceil \alpha k \rceil m+1)]$		$t[2\lceil \alpha k \rceil \ell + 2(h-1)k^3]$		$t[2\lceil \alpha k \rceil \ell + 2(i-1)k^3]$	
	$\downarrow$		$\downarrow$		$\downarrow$	
t_h:	... A C A C A	...	A C A C A C	...	C C C C C	...
t_i:	... T C T C T	...	C C C C C C	...	A C A C A	...
t_{k+h}:	... C C C C C	...	T C T C T C	...	C C C C C	...
t_{k+i}:	... C C C C C	...	C C C C C C	...	T C T C T	...

Let us define a "benchmark alignment" of the above sequences. We will define this alignment by partitioning the sequences into two sets $\mathcal{L}$ and $\mathcal{R}$ and inserting one space to the left of each sequence in $\mathcal{L}$ and one space to the right of each sequence in $\mathcal{R}$. Let $< V_1, V_2 >$ be a cut of G. We will show how to associate a benchmark alignment to such cut. For each $1 \le i \le k$ we let $t_i \in \mathcal{L}$ iff $t_{k+i} \in \mathcal{R}$. Moreover for each $1 \le i \le k$ we let $t_i \in \mathcal{L}$ iff $v_i \in V_1$.

Note that the score for the benchmark alignment is $4k^2 + \alpha k U$, where U is the number of edges that *are not* in the cut $< V_1, V_2 >$. Moreover, the benchmark alignment is a gap-0-1 alignment, and hence both a gap-0 alignment and a space-1 alignment.

We will show that there exists a fixed $\delta > 0$ such that if an alignment a of the above sequences is found that scores within a factor of $(1+\delta)$ of the benchmark alignment, then it will be possible to reconstruct, in polynomial time, from this alignment a partition of the vertex set that induces a cut whose size is within a additive constant of ϵk of maximum. Suppose we have *any* alignment a that scores within $1+\delta$ of our benchmark alignment, where δ is sufficiently small and will be determined later. Let us say that a sequence pair $< t_p, t_q >$ is *static in a* if there is no space in the induced pairwise alignment $< bt_p, bt_q >$. Being static in a is easily seen to be an equivalence relation. Let T_1 and T_2 denote the two largest equivalence classes of the "static" relation, and let T_3 denote the set of sequences that are neither in T_1 nor in T_2. Note that none of the sequence pairs $< t_i, t_{k+i} >$ can be static in a, otherwise the cost of the alignment of $< t_i, t_{k+i} >$ is too large. Thus the size of T_1 and T_2 is at most k. Let $|T_1| = k - k_1$, $|T_2| = k - k_2$. Then $|T_3| = k_1 + k_2$. Since each pair of sequences from different equivalence classes adds at least 4 to the SP-score of a, we have

$$SP(< at_1, \ldots, at_{2k} >) \geq 4((k-k_1)(k-k_2) + (k-k_1)(k_1+k_2) + (k-k_2)(k_1+k_2)) =$$

$$4(k^2 + k_1k_2 + k(k_1+k_2) - (k_1+k_2)^2) = 4(k^2 + k_1k_2 + (k-|T_3|)|T_3|).$$

Thus the numbers k_1 and k_2 must be such that $k_1k_2 + (k-|T_3|)|T_3| < \delta k^2 + \delta\alpha k U$, where U is the number of edges that are not cut by the partition used in the benchmark alignment. Note that $U \leq 3k$. We will choose $\delta < \frac{\epsilon\alpha}{100}$. It follows that as long as α is sufficiently small, we can assume that $|T_3| < k\frac{\epsilon}{6}$. Now let α, δ be as above, and let V_i be the set of all vertices such that $t_i \in T_i$ for $i \in \{1,2\}$. Consider the partition $< V_1, V \backslash V_1 >$. Let W be the number of edges of G that are not cut by $< V_1, V \backslash V_1 >$. Note that this number differs from the number Z of edges $\{v_i, v_j\}$ such that $< t_i, t_j >$ is static by at most $3|T_3|$, since every edge in the difference must have an endpoint in T_3 and the degree of the graph is 3. If the SP-score of the alignment is within a factor of $(1+\delta)$ of that of the benchmark alignment, then we have:

$$4k^2 + \alpha k W \leq 4k^2 + \alpha k(Z + k\frac{\epsilon}{2}) \leq (1+\delta)(4k^2 + \alpha k U) + \alpha\frac{\epsilon}{2}k^2.$$

By the choice of δ and since $U \leq 3k$, we get

$$\alpha k W - \alpha k U < 4\delta k^2 + \delta\alpha k U + \alpha\frac{\epsilon}{2}k^2.$$

Assuming, as we may, that $\alpha \leq 1$, and noting that $U \leq 3k$, our choice of δ gives:

$$W - U < 4\frac{\epsilon}{100}k + 3\frac{\epsilon}{100}\alpha k + \frac{\epsilon}{2}k < \epsilon k.$$

□

The following results on hardness of Switchboard Location problems are not covered by Theorem 3.

Theorem 4 *For every constant $\sigma > 1$, the* Switchboard Location$_2(\sigma)$ *problem is NP-hard.*

Proof. Let $\sigma > 1$. Since the number of instances of Switchboard Location$_2(\sigma)$ increases with σ, we may without loss of generality assume that $\sigma \leq 2$. We prove the theorem by reducing the Max-Cut problem to Switchboard Location$_2(\sigma)$. Given a graph $G =< V, E >$ with vertices $V = \{v_1, \ldots, v_k\}$, construct a metric space $X = \{x_1, \ldots, x_k, y_1, \ldots, y_k\}$ as follows: For $i \neq j$, we let $d(x_i, x_j) = d(y_i, y_j) = 1$. If $\{v_i, v_j\} \in E$, then $d(x_i, y_j) = \sigma$; if $\{v_i, v_j\} \notin E$, then $d(x_i, y_j) = 1$. (Note that for our choice of σ, the distance function is actually a metric.) For $1 \leq i \leq k$, the region R_i is defined as $\{x_i, y_i\}$. This gives us an instance I of the Switchboard Location$_2(\sigma)$ problem. Every solution $\bar{x}$ of I induces a partition $< V_x, V_y >$, where $V_x = \{v_i : x_i \in \bar{x}\}$ and $V_y = \{v_i : y_i \in \bar{x}\}$. If $c_{\bar{x}}$ denotes the size of the cut induced by the partition $< V_x, V_y >$, then the measure of $\bar{x}$ is equal to $\binom{k}{2} + (\sigma - 1)(|E| - c_{\bar{x}})$, and the theorem follows from NP-hardness of the Max-Cut problem (see [GJ79]). □

Theorem 5 *The* Switchboard Location$_2$ *problem is MAX SNP-hard.*

In view of our observation that Gap-0-1 Alignment is a special case of Switchboard Location, Theorem 5 is a corollary of Theorem 3(c) of [J99].

Acknowledgements

We thank Tao Jiang for helpful discussions about this paper and Arie Tamir for bringing references [AH97] and [T94] to our attention.

References

[AK97] P. Alimonti and V. Kann. Hardness of approximating problems on cubic graphs. *CIAC 97*, 288-298, volume 1203 of LNCS, 1997.

[AH97] E. M. Arkin and M. Hassin. Minimum diameter covering problems. *Technical report,* 1997.

[AKK99] S. Arora, D. Karger, and M. Karpinski. Polynomial Time Approximation Schemes for Dense Instances of $\mathcal{NP}$-Hard Problems. *J. Computer and Systems Sci.,* 58, 193-210, 1999.

[BCG93] S. A. Benner, M. A. Cohen, and G. H. Gonnet. Empirical and Structural Models for Insertions and Deletions in the Divergent Evolution of Proteins. *J. Mol. Biol.* 229:1065-1082, 1993.

[BLP97] V. Bafna, E.L. Lawler, P.A. Pevzner, Approximation algorithms for multiple sequence alignment, *Theoret. Comput. Sci.*, 182:233-244, 1997.

[BD00] P. Bonizzoni and G. Della Vedova. The complexity of multiple alignment with SP-score that is a metric. To appear in *Theoretical Computer Science.*

[F93] W. M. Fitch. Letter to the Editor: Commentary on the letter by Ward C. Wheeler. *Mol. Biol. Evol.* 10(3):713-714, 1993.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman 1979.

[G93] D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds, *Bull. Math. Biol.* 55:141-154, 1993.

[JKL98] T. Jiang, P. Kearney, and M. Li. Orchestrating Quartets: Approximation and Data Correction. *Proc. 39th IEEE Symp. on Found. of Comp. Science,* 416-425, 1998.

[JKL99] T. Jiang, P. Kearney, and M. Li. Some Open Problems in Computational Molecular Biology. *SIGACT News* 30(3):43-49, 1999.

[J99] W. Just. Computational Complexity of Multiple Sequence Alignment with SP-Score. *Submitted,* 1999.

[LMW99] M. Li, B. Ma, L. Wang, Finding Similar Regions in Many Strings, *Proc. 31st Symposium on the Theory of Computing (STOC)*, 473-482, 1999.

[PA92] S. Pascarella and P. Argos. Analysis of Insertions/Deletions in Protein Structures. *J. Mol. Biol.* 224:461-471, 1992.

[P92] P.A. Pevzner, Multiple alignment, communication cost, and graph matchings, *SIAM J. Appl. Math*, 52:1763-1779, 1992.

[T94] A. Tamir. A distance constrained p-facility location problem on the real line. *Mathematical Programming,* 66:201-204, 1994.

[WJ94] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology,* 1(4):337-348, 1994.

[W93] W. C. Wheeler. Letter to the Editor: The Triangle Inequality and character analysis. *Mol. Biol. Evol.* 10(3):707-712, 1993.