Multiple Alignment, Communication Cost, and Graph Matching

Pavel A. Pevzner

SIAM Journal on Applied Mathematics, Vol. 52, No. 6. (Dec., 1992), pp. 1763-1779.

Stable URL:
http://links.jstor.org/sici?sici=0036-1399%28199212%2952%3A6%3C1763%3AMACCAG%3E2.0.CO%3B2-Z

SIAM Journal on Applied Mathematics is currently published by Society for Industrial and Applied Mathematics.

SIAM J. APPL. MATH.
Vol. 52, No. 6, pp. 1763–1779, December 1992

015

MULTIPLE ALIGNMENT, COMMUNICATION COST, AND GRAPH MATCHING*

PAVEL A. PEVZNER†

Abstract. Multiple sequence alignment is an important problem in computational molecular biology. Dynamic programming for optimal multiple alignment requires too much time to be practical. Although many algorithms for suboptimal alignment have been suggested, no "performance guarantees" algorithms have been known until recently. A computationally efficient approximation multiple alignment algorithm with guaranteed error bounds equal to the normalized communication cost of a corresponding graph is given in this paper. Recently, Altschul and Lipman [*SIAM J. Appl. Math.*, 49 (1989), pp. 197–209] used suboptimal alignments for reducing the computational complexity of the optimal alignment algorithm. This paper develops the Altschul–Lipman approach and demonstrates that bounds for optimal multiple alignment of k sequences can be derived from a solution of the maximum weighted matching problem in a k-vertex graph. Fast maximum matching algorithms allow efficient implementation of dynamic bounds for the multiple alignment problem.

Key words. sequence alignment, biological sequences, design and analysis of algorithms, approximation algorithms, dynamic programming, maximum matching

AMS(MOS) subject classifications. 05C70, 68Q25, 92D20

1. Introduction.

Multiple sequence alignment is a difficult problem in computational molecular biology. The classical dynamic programming approach for the *optimal* multiple alignment problem reduces multiple alignment of k sequences of length n to the minimum path problem in the graph with n^k vertices representing a lattice in the k-dimensional space (see, for example, Sankoff [54], Waterman, Smith, and Beyer [76], Fredman [22], Murata, Richardson, and Sussman [49], Sankoff [55], Gotoh [27]). Since this approach is impractical for comparing more than three sequences, each with the length of an average protein, a number of algorithms for *suboptimal* multiple alignment have been developed.

Suboptimal algorithms can be separated into the following two groups (see recent reviews of Waterman [73], Argos, Vingron, and Vogt [7], Chan, Wong, and Chiu [14]):

(i) *Step-by-step pairwise alignment* (Waterman and Perlwitz [75]) with various strategies of clustering, iterative pairwise alignment with consensus (Bains [10]), or iterative dot-matrix comparison (Vingron and Argos [71]):

- Primary alignment of close sequences (Barton and Sternberg [11]) and clustering (Taylor [66], Corpet [18], Subbiah and Harrison [62]);
- Alignment according to phylogenetic trees (Hogeweg and Hesper [32], Feng and Doolittle [20], Hein [30]);
- Hierarchical sequence synthesis (Chan, Wong, and Chiu [14]);
- Alignment with post-processing (Tajima [63]).

(ii) *Local multiple alignment (consensus) methods* (Waterman, Arratia, and Galas [74], Johnson and Doolittle [35]) with local multiple alignments assembly (Sobel and Martinez [59], Waterman [72]):

- Consensus derived from consecutive pairwise comparisons (Bacon and Anderson [9]);

* Received by the editors September 3, 1991; accepted for publication (in revised form) January 13, 1992. The research of this author was supported in part by National Science Foundation grant DMS-90-05833 and National Institute of Health grant GM36230.

† Department of Mathematics, University of Southern California, Los Angeles, California 90089-1113.

- Assembly of weak consensuses (Santibanez and Rohde [56]);
- Fast local multiple alignment/consensus/motif search (Queen, Wegman, and Korn [52], Posfai et al. [51], Smith, Annau, and Chandrasegaran [58], Foulser and Core [21]) with estimations of statistical significance (Karlin et al. [36], Stormo and Hartzell [61], Lawrence and Reilly [39], Leung et al. [41], Schuler, Altschul, and Lipman [57]);
- Consensus search with clustering (Patthy [50], Martinez [48]);
- Phylogenetic tree consensus search (Higgins and Sharp [31]);
- Consensus assembly with post-processing (Vingron and Argos [70]);
- Consensus search and assembly with consistency checking of pairwise alignments (Gotoh [28]);
- Various priority criterions for consensus assembly (Chappey et al. [15]);
- Dot-matrix superimposing (Vihinen [69]);
- Dot-matrix projection overlapping (Roytberg [53]).

Recently, Carrillo and Lipman [13] suggested a new algorithm that combined both optimal and suboptimal alignment approaches and used suboptimal alignments for deriving bounds in optimal multiple alignment problems. They suggested a bounding procedure for constructing optimal multiple alignments allowing reduction of computational time and alignment of up to eight sequences (Lipman, Altschul, and Kececioglu [43]).

The bounds, suggested in Carrillo and Lipman [13], Spouge [60], and Altschul and Lipman [6], allow removal of regions in the k-dimensional lattice and reduction of the number of vertices in the corresponding graph to $V << n^k$. The value V (*computational volume*) depends on the suboptimal solution used for deriving bounds; the better the suboptimal solution, the smaller V is and therefore the computational time. Hence the problem of "good" suboptimal solution is a crucial issue in reducing the search for optimal multiple alignments.

A common approach in computer science to solve difficult optimization problems is to develop fast approximation algorithms whose maximum possible deviation from the optimal solution can be proved to be bounded by a small multiplicative constant c. Carrillo and Lipman [13] did not suggest an approximation algorithm for deriving "good" suboptimal solutions. Gusfield [29] first proposed an approximation algorithm for the multiple alignment problem with $c = 2 - 2/k$. It is known that models currently used to align sequences are not quite adequate (Lesk, Levitt, and Chothia [40], Taylor [65]); thus, for practical sequence alignment, it is not always necessary to produce an optimal alignment but only one that is plausible. The Gusfield [29] approximation algorithm produces plausible alignments; a computational experiment with an alignment of 19 sequences gave a suboptimal solution only 2 percent worse than the optimal solution.

This paper develops Gusfield's approximation algorithm and demonstrates that a better approximation could be derived from the solution of the *optimum communication spanning tree* problem (Hu [33]). Although this problem is NP-*complete* (Johnson, Lenstra, and Rinnooy Kan [34]), the *lower plane method* (Ahuja and Murty [2], [3]) is able to solve moderately large-sized problems ($k \approx 100$) in reasonable time. An approximation algorithm with $c = 2 - 3/k$ generalizing Gusfield's centered tree approach is suggested. Running time of the algorithm is defined by the running time of all triple alignments among k sequences and equals $O(n^3k^3 + k^4)$. We also formulate an open problem of devising polynomial approximation algorithms with $c = 2 - l/k$ for arbitrary fixed $l \leq k$.

The second part of the paper develops Altschul–Lipman [6] bounding procedures

for multiple alignment. Carrillo and Lipman [13] defined the cost of a multiple alignment to be the sum of the costs of all implied pairwise alignments. Recently, Altschul and Lipman [6] extended the Carrillo–Lipman algorithm to the definition of the multiple alignment score for the sequences $S_1S_2 \dots S_n$ as the cost of an evolutionary tree T having $S_1S_2 \dots S_n$ in the leaves. (See the pioneering paper of Sankoff [54].) This definition is in closer agreement with biological intuition.

The Altschul–Lipman algorithm is based on the study of a polyhedron describing paths between leaves of T. Let $G(V, E)$ be an undirected graph, $N \subset V$ be an arbitrary set of *poles* in G, and $\mathcal{P}_N$ be the set of all paths in G joining the vertices from N. The *packing problem* (Garey and Johnson [26]) for family $\mathcal{P}_N$ is known as a *free multiflow problem* and has been intensively studied in combinatorial optimization (Adel'son-Velsky, Dinic, and Karzanov [1], Lomonosov [44]). In fact, the Altschul–Lipman polyhedron is a very particular case of the polyhedron of the free multiflow problem for the case when G is a tree and N is the set of all leaves of this tree. As clarified by Lovasz [45] and Cherkassky [17], the free multiflow problem has a half-integer optimal solution for an arbitrary graph G and an arbitrary set of poles (compare with Lemma 1 of Altschul and Lipman [6]).

In the case when the tree T is a star, Altschul and Lipman [6] suggested solving a fractional programming problem for evolutionary tree multiple alignment and raised a problem of devising polynomial-time algorithms for deriving bounds for alignment. For stars with a small number of edges, they even enumerate all vertices of the corresponding polyhedron. In this paper, a combinatorial approach is suggested that avoids using fractional programming and enumerating of the polyhedron's vertices. It is shown that the problem of finding a maximum free multiflow with an arbitrary objective function is reduced to a *fractional weighting matching* problem (Lovasz and Plummer [46]) in the case when T is a star. Characterization of fractional matching polytopes (Balinski [8]) gives a complete treatment of the problem of finding bounds in the Altschul–Lipman algorithm (see Theorem 1 of Altschul and Lipman [6]). Reducing this to a maximum fractional weighting matching problem (Lovasz and Plummer [46]) allows implementation of efficient bounding procedures for multiple alignment.

The paper is organized as follows. In §2 we formulate the multiple alignment problem as a shortest path problem in an alignment graph. In §3 we discuss various approaches to the definition of multiple alignment score. In §4 we introduce l-stars and give a generalization of Feng–Doolittle construction for multiple alignment consistent with a tree. In §5 we define the communication cost of graphs and give a few examples of communication cost calculations. In §6 we establish connections between the communication cost of graphs and guaranteed error bounds for multiple alignment problems. In §7 we consider the minimum communication spanning tree problem and give a multiple alignment algorithm with guaranteed error bound $2 - 3/k$. In §8 we discuss the dynamic upper bounds for evolutionary tree alignment in terms of linear programming. In §9 we use fractional graph matching for deriving dynamic upper bounds for evolutionary tree alignment.

2. Alignment graph. Let $\mathcal{A}$ be an alphabet of α letters ($\alpha = 4$ for nucleotide sequences and $\alpha = 20$ for amino acids sequences). Denote $\sum = \mathcal{A} \bigcup \Delta$; Δ is often said to represent an *indel* or *in*sertion/*del*etion of a letter. Let $S_1 = s_1(1)s_1(2) \dots s_1(l_1)$, $S_2 = s_2(1)s_2(2) \dots s_2(l_2)$, $S_k = s_k(1)s_k(2) \dots s_k(l_k)$ be a set of sequences over the alphabet $\mathcal{A}$ of length $l_1, l_2, \dots, l_k$, respectively.

Consider a set of k-dimensional integer vectors $V = \{\mathbf{v} : \ \mathbf{0} \leq \mathbf{v} \leq \ell\}$ and define

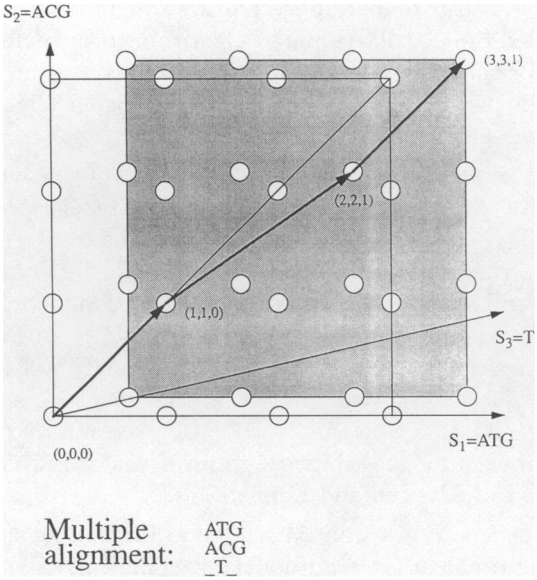

FIG. 1. *A path* $((0,0,0),(1,1,0),(2,2,1),(3,3,2))$ *from* $(0,0,0)$ *to* $(3,3,2)$ *in alignment graph, consisting of three arcs, corresponds to the triple alignment of sequences* S_1 S_2 S_3 *with three columns.*

the *alignment digraph* $G(V,E)$ with the arc set E by the rule

$$(\mathbf{v},\mathbf{v}')\epsilon E \Longleftrightarrow \mathbf{0} < \mathbf{v}' - \mathbf{v} \leq \mathbf{1}$$

(here $\ell = (l_1 l_2 \ldots l_k)$, $\mathbf{0} = (00\ldots0)$, $\mathbf{1} = (11\ldots1)$). Denote by μ the set of all *paths* in G from $\mathbf{0}$ to ℓ.

Each arc e in G between $\mathbf{v} = (v_1 v_2 \ldots v_k)$ and $\mathbf{v}' = (v_1' v_2' \ldots v_k')$ corresponds to the k-tuple $f(e) = (a_1 a_2 \ldots a_k)$ in the alphabet $\sum$ according to the rule

$$a_i = \begin{cases} s_i(v_i') & \text{if } v_i < v_i', \\ \Delta & \text{otherwise.} \end{cases}$$

Each path $(e_1 e_2 \ldots e_n)\epsilon\mu$ can be represented as a $k \times n$ matrix, where the ith column of this matrix corresponds to $f(e_i)$. This representation is known as a *multiple alignment* of $S_1 S_2 \ldots S_k$, and n is referred to as the number of columns of the multiple alignment (Fig. 1). The sequences $S_1 S_2 \ldots S_k$ can be used for defining the lengths on μ; in this case, the *minimum path problem* in G is referred to as the *optimal* (*multiple*) *alignment problem.*

Let $(d(a_i,a_j))$ be a $|\sum| \times |\sum|$ two-dimensional *distance* matrix. If both $a_i, a_j \neq \Delta$, $d(a_i,a_j)$ is the weight of the *substitution* a_j for a_i; if $a_i = \Delta$, $d(\Delta,a_j)$ is the weight of *insertion* a_j; if $a_j = \Delta$, $d(a_i,\Delta)$ is the weight of *deletion* a_i. (See Altschul [5] for a discussion of various matrices for comparing DNA and protein sequences.) For the case of two sequences S_1 and S_2, the distance matrix can be used for defining the arc lengths in G

$$d(e) = d(f(e)) = d(a_1,a_2),$$

and the length of the path $(e_1 e_2 \ldots e_n)\epsilon\mu$ can be defined simply as the sum of the

lengths of its arcs as

$$d(e_1e_2\dots e_n)=\sum_{i=1}^{n} d(e_i). \tag{1}$$

3. The score of multiple alignment. To generalize the last definition for $k>2$ sequences, we can define a k-dimensional distance matrix $d(a_1a_2\dots a_k)$ and assume that

$$d(e)=d(f(e))=d(a_1a_2\dots a_k). \tag{2}$$

For example, the k-dimensional matrix defined by the rule

$$d(e)=d(a_1a_2\dots a_k)=\begin{cases} 1 & \exists a\epsilon\mathcal{A}: \quad a_i=a \text{ or } a_i=\Delta \quad \text{for all } i\epsilon\{1,2,\dots,k\},\\ \infty & \text{otherwise}\end{cases}$$

corresponds to the *multiple shortest common supersequence* (MSCS) problem. The decision version of MSCS was shown to be NP-complete for alphabet size $\alpha\geq 5$ (Maier [47]). Timkovsky [67] proved the NP-complexity of the MSCS for a few particular cases and suggested an approximation algorithm without guaranteed bounds.

The k-dimensional matrix defined by the rule

$$d(e)=d(a_1a_2\dots a_k)=\begin{cases} -1 & \text{if } a_1=a_2=\ldots=a_k\neq\Delta,\\ 0 & \text{otherwise}\end{cases}$$

corresponds to the *multiple longest common subsequence* (MLCS) problem. Maier [47] proved that the decision version of MLCS is NP-complete. Timkovsky [67] raised the open problems of devising the efficient approximation algorithms for MSCS and MLCS with guaranted error bounds.

In the biological applications, $d(a_1a_2\dots a_k)$ is often defined through the two-dimensional matrix by

$$d(a_1a_2\dots a_k)=\sum_{1\leq i<j\leq k} d(a_i,a_j). \tag{3}$$

In this case, the length of the path $P\epsilon\mu$ equals the sum of the lengths of *projections* of P onto all pairs of sequences S_i and S_j. This function is called sum-of-the-pairs or SP-*score*. Another definition of alignment score will be considered in §8; see Altschul [4] for various approaches to the definition of alignment score. A natural generalization of (3) is the *weighted* SP-*score*

$$d(a_1a_2\dots a_k)=\sum_{1\leq i<j\leq k} c_{i,j}\cdot d(a_i,a_j), \tag{4}$$

where $c_{i,j}$ is the "weight" of the pairwise alignment of S_i and S_j.

In this paper, we assume that $d(\Delta,\Delta)=0$ and that d satisfies the *triangle inequality*

$$\forall a_1,a_2,a_3: \quad d(a_1,a_3)\leq d(a_1,a_2)+d(a_2,a_3). \tag{5}$$

Comment. The alignment graph with the lengths of the arcs defined by the rule

$$d(e)=d(\mathbf{v},\mathbf{v}',a_1,a_2,\dots,a_k)=d((v_1,v_2,\dots,v_k),(v_1',v_2',\dots,v_k'),a_1,a_2,\dots,a_k)$$

$$= \begin{cases} \infty & \text{if } \exists i: \quad 0 < v_i = v_i' < l_i \text{ or } \exists i,j: \quad a_i, a_j \neq \Delta \text{ and } a_i \neq a_j, \\ 1 & \text{otherwise} \end{cases}$$

corresponds to the *multiple shortest common superstring* problem. The multiple shortest common superstring problem was shown to be NP-complete by Gallant, Maier, and Storer [25]. Tarhio and Ukkonen [64], Turner [68], and Timkovsky [67] raised the problem of devising the efficient approximation algorithms with guaranteed error bounds for the multiple shortest common superstring. This problem has been solved by Li [42] and Blum et al. [12].

Comment. It is quite common in computational molecular biology to refer to multiple alignment as an NP-complete problem (Eppstein et al. [19]). It is worth noting that questions about the computational complexity of the optimal alignment with SP-score or evolutionary tree score are still open. (Maier's [47] reduction of a vertex cover problem to MLCS is not generalized for these scores.)

4. Configurations and *l*-stars. In this section, we generalize the Feng and Doolittle [20] construction for multiple alignment consistent with a tree (for definitions from graph theory and linear programming, see Lovasz and Plummer [46]).

Denote by $[1:k]$ the set of integers $1 \leq i \leq k$. Given a set of k sequences $S_1 S_2 \ldots S_k$ and $\Omega = \{i_1 i_2 \ldots i_t\} \subset [1:k]$, we denote by $D^{\text{opt}}(\Omega)$ ($D^{\text{opt}}(S_{i_1}, S_{i_2}, \ldots, S_{i_t})$) the score of optimal alignment of $S_{i_1}, S_{i_2}, \ldots, S_{i_t}$. In particular, $D^{\text{opt}}(S_i, S_j)$ denotes the score of the optimal pairwise alignment of S_i and S_j, while $D^{\text{opt}}(S_i, S_j, S_k)$ denotes the score of the optimal triple alignment of S_i, S_j, and S_k. Given a multiple alignment A, we denote by $D(A)$ the score of alignment A and, by $D(A|\Omega)$, the score of the multiple alignment of $S_{i_1}, S_{i_2}, \ldots, S_{i_t}$ *induced* by A. Obviously, for an arbitrary alignment, $D(A|\Omega) \geq D^{\text{opt}}(\Omega)$. A is called Ω-*consistent* if $D(A|\Omega) = D^{\text{opt}}(\Omega)$.

Let $V = [1:k]$, $G(V, E)$ be an *undirected graph* and $\{\Omega_1, \Omega_2, \ldots, \Omega_t\}$ be the list of all *cliques* of G. Denote $W_1 = [1:t]$, $W_2 = \{v: \quad v \epsilon \Omega_i \bigcap \Omega_j \text{ for } 1 \leq i < j \leq t\}$. Define a *bipartite* graph $G^c(W_1 \bigcup W_2, E^c)$ with the parts W_1 and W_2 and the edge set E^c (Fig. 2)

$$(i, v) \epsilon E^c \Longleftrightarrow v \epsilon \Omega_i \bigcap \Omega_k \quad \text{for a set } \Omega_k.$$

The graph G is called a *configuration* if G^c is a tree. (It implies that $|\Omega_i \bigcap \Omega_j| \leq 1$ for $1 \leq i < j \leq t$.)

A configuration fulfilling the conditions

(i) $|\Omega_1| = |\Omega_2| \cdots = |\Omega_t| = l$,

(ii) $\Omega_i \bigcap \Omega_j = \{x\}$ for $1 \leq i < j \leq t$

is called *l-star*, and x is called the *center* of the l-star (Fig. 3(a)).

Feng and Doolittle [20] observed that, given any tree T, where each vertex is labeled with a distinct sequence, there is a multiple alignment A of these sequences that is consistent with the optimal pairwise alignment corresponding to the edges of T. That is, if S_i and S_j are sequences corresponding to any two adjacent vertices of T, then the pairwise alignment of S_i and S_j induced by A has score exactly $D^{\text{opt}}(S_i, S_j)$.

The following lemma generalizes the Feng and Doolittle [20] construction (recall that a multiple alignment A is consistent with a p-vertex clique Ω if the multiple alignment of p sequences from Ω induced by A has score exactly $D^{\text{opt}}(\Omega)$).

LEMMA 1. *If G is a configuration with the cliques $\Omega_1, \Omega_2, \ldots, \Omega_t$, then there exist a multiple alignment of the sequences $S_1, S_2, \ldots, S_k$ that is consistent with Ω_1, $\Omega_2, \ldots, \Omega_t$.*

For the case where $|\Omega_1| = |\Omega_2| \ldots = |\Omega_t| = 2$ (G is a tree), the lemma yields the Feng–Doolittle construction for the multiple alignment consistent with the tree.

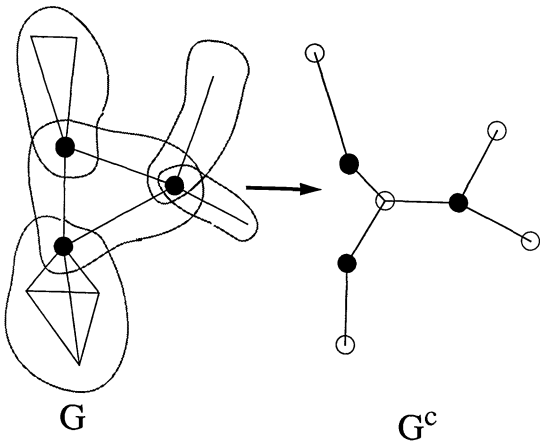

FIG. 2. *Example of graph (configuration) G with five cliques* ($|W_1| = 5, |W_2| = 3$, *the set* W_2 *is represented by black circles). Clique intersections of graph G determine the bipartite graph (tree)* G^c.

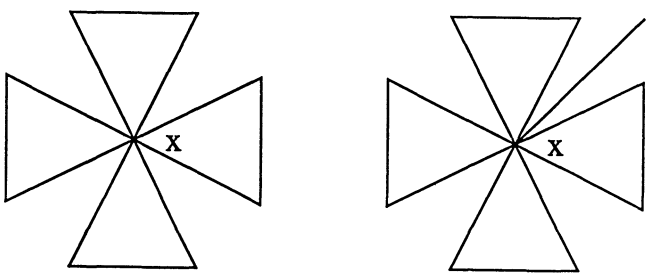

FIG. 3. *Examples of* (a) 3*-star and* (b) 3*-star with an additional edge.*

5. Communication cost. For a connected graph $G(V, E)$, define the *communication cost* $c(G)$ as

$$c(G) = \sum_{i,j \epsilon V} l(i,j),$$

where $l(i,j)$ denotes the number of edges in the shortest path between i and j in G, and the sum is taken over all unordered pairs of distinct vertices i and j. This definition generalizes *communication spanning tree cost* (Hu [33]). Fix a shortest path $\gamma(i,j)$ for each (unordered) pair of distinct vertices $i, j \epsilon V$ and denote by $\Gamma(G|e)$ the number of paths $\gamma(i,j)$ containing the edge e. Note that

$$c(G) = \sum_{e \epsilon E} \Gamma(G|e). \tag{6}$$

The *complete graph* H_k has minimum communication cost among all k-vertices graphs. We call $b(G) = c(G)/c(H_k) = 2(c(G)/k(k-1))$ the *normalized communication cost* of G.

Examples.

1. For each 2-star with $k = t + 1$ vertices, $c(G) = t^2$ and $b(G) = 2 - 2/k$.
2. For each 3-star with $k = 2t + 1$ vertices, $c(G) = (2t - 1)2t + t$ and $b(G) = 2 - 3/k$.
3. For the configuration of Fig. 3(b) with $k = 2t + 2$ edges (3-star with an additional edge), $c(G) = 2t \cdot 2t + t + (2t + 1) = 4t^2 + 3t + 1$ and $b(G) = 2 - 3/k + 2/(k(k-1))$.
4. For each l-star with $k = (l-1)t + 1$ vertices, $c(G) = ((l-1)t + 1 - l + 1)(l - 1)t + t((l(l-1))/(2) - l + 1)$ and $b(G) = 2 - l/k$.

6. Guaranteed error bounds for multiple alignment. In this section, for an arbitrary configuration G, we construct an alignment with guaranteed error bounds equal to the normalized communication cost of G.

Let $G(V, E)$ be a configuration and let $H_k(V, E_k)$ be a complete (undirected) graph with k vertices (we assume that $V = [1 : k]$). Let $\mathcal{G}$ be the set of all subgraphs of H_k *isomorphic* to G. For $G'(V, E')\epsilon\mathcal{G}$, we denote by $\omega' : V \longrightarrow V$ the *isomorphism* transforming G into G'. Correspondingly, the image of the edge $(i, j)\epsilon E$ under isomorphism ω' will be $(\omega'(i), \omega'(j))\epsilon E'$.

For a configuration $G'(V, E')\epsilon\mathcal{G}$ and an alignment A, denote

$$F(A|G') = \sum_{(i,j)\epsilon E'} \Gamma(G'|(i, j)) \cdot D(A|S_i, S_j). \tag{7}$$

(We assume that the family of shortest paths $(\gamma'_{i,j})$ in G' is induced (through ω') by the family of shortest paths $(\gamma_{i,j})$ in G).

LEMMA 2. *For an arbitrary alignment A and configuration G, $F(A|G) \geq D(A)$.*

Proof. Recall that $D(A|S_i, S_j)$ is the score of pairwise alignment S_i and S_j induced by A, and $D(A)$ is the SP-score of A. Due to (1), (3), and (5), $D(A|S_i, S_j)$ is a metric on V; therefore

$$F(A|G) = \sum_{(i,j)\epsilon E} \Gamma(G|(i, j)) \cdot D(A|S_i, S_j) = \sum_{(i,j)\epsilon E} \left(\sum_{l,m\epsilon V,\ (i,j)\epsilon\gamma(l,m)} D(A|S_i, S_j) \right)$$

$$= \sum_{l,m\epsilon V} \left(\sum_{(i,j)\epsilon\gamma(l,m)} D(A|S_i, S_j) \right) \geq \sum_{l,m\epsilon V} D(A|S_l, S_m) = D(A)$$

(the sums are taken over all unordered pairs of distinct vertices $l, m\epsilon V$). □

LEMMA 3. *For an arbitrary alignment A and an edge $(i, j)\epsilon E$,*

$$\sum_{G'\epsilon\mathcal{G}} D(A|S_{\omega'(i)}, S_{\omega'(j)}) = \frac{|\mathcal{G}|}{|E_k|} D(A).$$

Proof. Let $e = (l, k)\epsilon E_k$ be an arbitrary edge of E_k. Consider the set $\mathcal{G}_e = \{G'\epsilon\mathcal{G} : (w'(i), w'(j)) = (l, k)\}$. Due to symmetry, $|\mathcal{G}_e| = |\mathcal{G}|/|E_k|$ and

$$\sum_{G'\epsilon\mathcal{G}} D(A|S_{\omega'(i)}, S_{\omega'(j)}) = \sum_{e\epsilon E_k} \Big(\sum_{G'\epsilon\mathcal{G}_e} D(A|S_{\omega'(i)}, S_{\omega'(j)}) \Big)$$

$$= \sum_{e\epsilon E_k} \Big(\sum_{G'\epsilon\mathcal{G}_e} D(A|S_l, S_k) \Big)$$

$$= \sum_{G'\epsilon\mathcal{G}_e} \Big(\sum_{e\epsilon E_k} D(A|S_l, S_k) \Big) = \frac{|\mathcal{G}|}{|E_k|} D(A). \qquad \square$$

Now we introduce the notion of an optimal G-configuration. For a fixed $G' \epsilon \mathcal{G}$, let $A(G')$ be an alignment satisfying

$$F(A(G')|G') = \min_{A} F(A|G') \tag{8}$$

and let $G^* \epsilon \mathcal{G}$ (*optimal G-configuration*) be a configuration satisfying

$$F(A(G^*)|G^*) = \min_{G' \epsilon \mathcal{G}} F(A(G')|G'). \tag{9}$$

For simplicity, denote $A^* = A(G^*)$. Let A^{opt} be the optimal alignment of $S_1, S_2, \ldots, S_k$.

THEOREM 1. *For an arbitrary configuration G, the normalized communication cost $b(G)$ is a guaranteed error bound for alignment A^*,*

$$\frac{D(A^*)}{D(A^{\text{opt}})} \leq b(G).$$

Proof. Let M be the number of configurations in $\mathcal{G}$ containing the edge e of E_k. Note that, due to symmetry,

$$|\mathcal{G}|/M = |E_k|/|E|. \tag{10}$$

Define

$$W = \frac{1}{M} \cdot \sum_{G' \epsilon \mathcal{G}} F(A^{\text{opt}}|G').$$

According to (8) and (9),

$$W = \frac{1}{M} \cdot \sum_{G' \epsilon \mathcal{G}} F(A^{\text{opt}}|G') \geq \frac{1}{M} \cdot \sum_{G' \epsilon \mathcal{G}} F(A(G')|G') \geq \frac{|\mathcal{G}|}{M} \cdot F(A^*|G^*).$$

Therefore, by (10) and Lemma 2,

$$W \geq \frac{|E_k|}{|E|} D(A^*). \tag{11}$$

On the other hand,

$$\begin{aligned} W &= \frac{1}{M} \sum_{G'(V,E') \epsilon \mathcal{G}} \Big(\sum_{(i,j) \epsilon E'} \Gamma(G'|(i,j)) \cdot D(A^{\text{opt}}|S_i, S_j) \Big) \\ &= \frac{1}{M} \sum_{G' \epsilon \mathcal{G}} \Big(\sum_{(l,m) \epsilon E} \Gamma(G'|\omega'(l), \omega'(m)) \cdot D(A^{\text{opt}}|S_{\omega'(l)}, S_{\omega'(m)}) \Big). \end{aligned}$$

As the family of shortest paths $(\gamma'_{i,j})$ in G' is induced through ω' by the family of shortest paths $(\gamma_{i,j})$ in G, we have that $\Gamma(G'|\omega'(l), \omega'(m)) = \Gamma(G|l,m)$. Therefore, according to Lemma 3 and (6),

$$\begin{aligned} W &= \frac{1}{M} \sum_{(l,m) \epsilon E} \Gamma(G|l,m) \Big(\sum_{G' \epsilon \mathcal{G}} D(A^{\text{opt}}|S_{\omega'(l)}, S_{\omega'(m)}) \Big) \\ &= \frac{1}{M} \cdot \frac{|\mathcal{G}|}{|E_k|} D(A^{\text{opt}}) \cdot \sum_{(l,m) \epsilon E} \Gamma(G|l,m) \\ &= \frac{1}{M} \cdot \frac{|\mathcal{G}|}{|E_k|} D(A^{\text{opt}}) \cdot c(G). \end{aligned}$$

According to (10),

$$W = \frac{1}{|E|} D(A^{\text{opt}}) \cdot c(G),$$

and therefore, according to inequality (11),

$$\frac{D(A^*)}{D(A^{\text{opt}})} \leq \frac{c(G)}{|E_k|} = b(G). \qquad \square$$

COROLLARY 1 (Gusfield [29]). *If G is 2-star with k vertices, then*

$$\frac{D(A^*)}{D(A^{\text{opt}})} \leq 2 - \frac{2}{k}.$$

COROLLARY 2. *If G is l-star with k vertices, then*

$$\frac{D(A^*)}{D(A^{\text{opt}})} \leq 2 - \frac{l}{k}.$$

COROLLARY 3. *If G is 3-star with an additional edge having k vertices, then*

$$\frac{D(A^*)}{D(A^{\text{opt}})} \leq 2 - \frac{3}{k} + \frac{2}{k(k-1)}.$$

7. Search for an optimal 3-star. Theorem 1 reduces the problem of devising an approximation algorithm with guaranteed error bound to the search for an optimal G-configuration. In this section, we give a polynomial algorithm to find an optimal G-configuration for a 3-star G (*optimal 3-star*).

Let $H_k(V, E_k, D)$ be the complete weighted graph with the weights $D(i,j) = D^{\text{opt}}(S_i, S_j)$. Gusfield [29] defined a *center star* to be a 2-star of minimum weight in $H_k(V, E_k, D)$. For the case when G is a 2-star with $k = t+1$ vertices, $\Gamma(G|(i,j)) = t$ for each edge (i,j). Therefore, a center star is an optimal configuration for 2-star (it yields the minimum in (9) and gives an upper bound for the score of A^* equal to $t\sum_{i \neq x} D(S_x, S_i)$, where x denotes the center of the center star).

The computation of the weight function D needs $O(n^2 \cdot k^2)$ operations. According to Corollary 1, a center-star method gives an $O(n^2 \cdot k^2)$ algorithm for a multiple alignment problem with guaranteed upper bound $c = 2 - 2/k$. Note that a *minimum communication spanning tree* (Hu [33]) in $H_k(V, E_k, D)$ gives, in general, better alignment than the centered tree, especially in the case when among k sequences there exist triples S_i, S_j, S_l such that S_i is an ancestor of S_j, and S_j is an ancestor of S_k (see Hu [33, Thm. 3]). It can be proved also that the Waterman and Perlwitz [75] "line geometry" algorithm for constructing multiple alignments gives, in general, a better alignment than the multiple alignment consistent with a tree if the order of pairwise alignments in the Waterman–Perlwitz algorithm corresponds to the tree.

Next, we establish the following result: If G is 3-star with $2t+1$ vertices, an optimal 3-star G^* and an alignment A^* yielding the minimum in (9) can be found in time $O(n^3 \cdot k^3 + k^4)$. Consider the set of graphs $\mathcal{G}_x = \{G' : G' \text{ is a 3-star with center } x\}$. Let G_x^* be a graph satisfying

$$F(A(G_x^*)|G_x^*) = \min_{G' \epsilon \mathcal{G}_x} F(A(G')|G'). \tag{12}$$

Note that $\mathcal{G} = \bigcup_{x \epsilon V} \mathcal{G}_x$, and therefore

$$F(A(G^*)|G^*) = \min_{x \epsilon V} F(A(G_x^*)|G_x^*). \tag{13}$$

To find a configuration G_x^*, consider the weighted complete graph $H_{k-1}(V \setminus \{x\}, E_{k-1}, w)$ with $k-1$ vertices. The weight function $w(i,j)$ is defined as the score of a triple alignment A for sequences S_i, S_j, S_x minimizing

$$w(i,j) = D(A|S_i, S_j) + (k-2)(D(A|S_i, S_x) + D(A|S_j, S_x)). \tag{14}$$

The alignment A can be found in $O(n^3)$ time as a triple weighted SP-alignment according to (4).

THEOREM 2. *Let $(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)$ be a perfect matching of minimum weight in H_{k-1}. Let $G_x^*(V, E_x^*)$ be the 3-star defined by the edge set*

$$E_x^* = \{(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t),\ (i_1, x), (i_2, x), \ldots, (i_t, x),\ (j_1, x), (j_2, x), \ldots, (j_t, x)\}.$$

Then $G_x^*(V, E_x^*)$ yields a minimum in (12),

$$F(A(G_x^*)|G_x^*) = \min_{G' \epsilon \mathcal{G}_x} F(A(G')|G').$$

Proof. Let $G(V, E)$ be a 3-star with center x. Note that, for each edge $(x,i) \epsilon E$, $\Gamma(G|(x,i)) = k-2$, and, for each edge $(i,j) \epsilon E$ with $i, j \neq x$, $\Gamma(G|(i,j)) = 1$ (Fig. 3(a)). Therefore,

$$\begin{aligned} F(A|G) &= \sum_{(i,j) \epsilon E,\ \ i,j \neq x} D(A|S_i, S_j) + (k-2)(D(A|S_i, S_x) + D(A|S_j, S_x)) \\ &= \sum_{(i,j) \epsilon E,\ \ i,j \neq x} w(i,j). \end{aligned}$$

The last equation implies that the value $F(A|G)$ equals the score of perfect matching in H_{k-1} defined by edges $\{(i,j) \epsilon E : \ \ i, j \neq x\}$. □

Applying an algorithm for the weighted matching problem (Gabow [23], Galil [24]), this theorem implies an $O(n^3 \cdot k^3 + k^4)$ approximation algorithm for multiple alignment with guaranteed bound $c = 2 - 3/k$ (for odd k). The approximation algorithm for even k with $c = 2 - 3/k + 2/k(k-1)$ can be also implemented with $O(n^3 \cdot k^3 + k^4)$ running time.

Although Corollary 2 raises the possibility of devising an approximation algorithm based on l-sequence alignments with $c = 2 - l/k$, we do not know of a polynomial algorithm for optimal l-star search for $l > 3$.

Conjecture. For an arbitrary fixed l, there exists a polynomial approximation algorithm for multiple alignment of $k \geq l$ sequences with guaranteed upper bound $c \leq 2 - l/k$.

8. Bounds for multiple-tree alignment. While the SP-score is a simple measure on k-tuples, it has no clear foundation in the theory of molecular evolution. Sankoff [54] takes an approach in closer agreement with biological intuition. He assumes an *evolutionary tree* $T(V \bigcup W, E)$ with k leaves V and p internal vertices W. Input sequences $S_1 S_2 \ldots S_k$ are assigned to the leaves, and the additional *reconstructed* sequences $S_{k+1} S_{k+2} \ldots S_{k+r}$ are assigned to the internal vertices. In this model, the

letters $a_{k+1}, a_{k+2}, \ldots a_{k+r}$ are assigned to p internal vertices of the tree, and the weights of the k-tuples are defined by the rule

$$d(a_1, a_2, \ldots, a_k) = \min_{a_{k+1}, a_{k+2}, \ldots, a_{k+r}} \sum_{(i,j)\epsilon E} d(a_i, a_j). \tag{15}$$

The score of multiple alignments defined by (1), (2), and (15) is the *tree score*. Obviously, the tree score $D(A)$ equals the sum of pairwise alignment scores defined by the edges of T. The special case when the tree has only one internal node is a *star alignment* with *star score*.

The Altschul and Lipman [6] algorithm for finding an optimal tree alignment has the following three steps:

1. Find an upper bound on the score of each projection of an optimal alignment $D(A^{\text{opt}}|S_i, S_j)$;
2. Use these bounds to reduce the computational volume of the alignment graph;
3. Find an optimal path within the reduced alignment graph.

We begin by summarizing part (1) of the Altschul–Lipman algorithm. Set $P = \{(i,j) : 1 \leq i < j \leq k\}$ and let $\rho(i,j)$ be the edge set of the path between i and j in T. Define the function $\delta : P \times E \longrightarrow \{0,1\}$ by

$$\delta((i,j)|e) = \begin{cases} 1 & \text{if the path between } i \text{ and } j \text{ in } T \text{ passes through edge } e \text{ of the tree,} \\ 0 & \text{otherwise.} \end{cases}$$

Let A be an alignment of $S_1 S_2 \ldots S_k$ with the tree score $D(A) = \sum_{(i,j)\epsilon E} D(A|S_i, S_j)$. Assume that, for each $(i,j)\epsilon P$, we know some *lower bound* on $D(A|S_i, S_j)$

$$D(A|S_i, S_j) \geq C_{i,j}. \tag{16}$$

As $D(A|S_i, S_j)$ is a metric on $V \bigcup W$, the lower bounds (16) imply the inequalities

$$\forall (i,j)\epsilon P : \quad \sum_{(l,m)\epsilon\rho(i,j)} D(A|S_l, S_m) \geq D(A|S_i, S_j) \geq C_{i,j}. \tag{17}$$

Therefore, the optimal solution of the following *linear program* with the variables $D(A|S_l, S_m)$:

$$\begin{aligned} &\forall (l,m)\epsilon E : \quad D(A|S_l, S_m) \geq 0, \\ &\forall (i,j)\epsilon P : \quad \sum_{(l,m)\epsilon E} \delta((i,j)|(l,m)) \cdot D(A|S_l, S_m) \geq C_{i,j}, \\ &\qquad\qquad \min \left\{ \sum_{(i,j)\epsilon E} D(A|S_i, S_j) \right\}, \end{aligned} \tag{18}$$

implies a lower bound for $D(A)$.

Due to the *duality theorem* of linear programming, the value of the optimal solution of (18) equals the value of the optimal solution of the linear program

$$\begin{aligned} &\forall (i,j)\epsilon P : \quad x_{i,j} \geq 0, \\ &\forall e\epsilon E : \quad \sum_{(i,j)\epsilon P} \delta((i,j)|e) \cdot x_{i,j} \leq 1, \\ &\qquad\qquad \max \left\{ \sum_{(i,j)\epsilon P} C_{i,j} x_{i,j} \right\}. \end{aligned} \tag{19}$$

Therefore, each solution $(x_{i,j})$ of (19) implies a lower bound for $D(A)$,

$$D(A) \geq \sum_{p \epsilon P} C_p x_p. \tag{20}$$

Fix $q \epsilon P$ and a solution $(x_{i,j})$ of (19). If $x_q > 0$, then (20) implies that

$$\left(D(A) - \sum_{p \epsilon P, p \neq q} C_p x_p \right) / x_q \geq C_q.$$

If we had an upper bound $C' \geq D(A)$ for $D(A)$, it would imply an upper bound for C_q,

$$\left(C' - \sum_{p \epsilon P, p \neq q} C_p x_p \right) / x_q \geq C_q.$$

The linear program (19) can be used to reduce the computational volume of the alignment graph. Let $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ be an arbitrary vertex of alignment graph and let $A(\mathbf{v})$ be an alignment of minimal tree score among all alignments passing through $\mathbf{v}$. Let $C_{i,j}(\mathbf{v})$ be the optimal pairwise alignment among all pairwise alignments of S_i, S_j passing through (v_i, v_j). Obviously, $C_{i,j}(\mathbf{v})$ is a lower bound (16) for $D(A(\mathbf{v})|S_i, S_j)$, and therefore the optimal solution of (19) with coefficients $C_{i,j} = C_{i,j}(\mathbf{v})$ implies a lower bound $C(\mathbf{v})$ for $A(\mathbf{v})$. If $C(\mathbf{v}) > C'$, the vertex $\mathbf{v}$ can be excluded from computational volume of the alignment graph. Following Spouge [60], we call the upper bound $C(\mathbf{v})$ a *dynamic* upper bound. (These upper bounds can be calculated upon a minimum path search in the alignment graph.)

Altschul and Lipman [6] suggested using the solution of *fractional program*

$$\begin{aligned} & & x_q &> 0, \\ & \forall p \epsilon P & x_p &\geq 0, \\ & \forall e \epsilon E & \sum_{e \epsilon E} \delta(p|e) \cdot x_p &\leq 1, \end{aligned} \tag{21}$$

$$\max \left\{ \left(C' - \sum_{p \epsilon P, p \neq q} C_p x_p \right) / x_q \right\}$$

for deriving a *static* upper bound on C_q. For stars with a small number of edges, they even enumerate all vertices of the corresponding polyhedron. In this paper, we suggest a combinatorial approach for deriving both dynamic and static upper bounds by reducing (19) and (21) to fractional weighted matching problems (Lovasz and Plummer [46]) in the case when T is a star.

Comment. Altschul and Lipman [6] raised the problem of devising a polynomial algorithm for (21). Note that a simple reduction of (21) to linear program (Charnes and Cooper [16]) allows application of Khachian [38] and Karmarkar [37] polynomial algorithms for fractional programming.

9. Multiple star alignment and fractional graph matchings. In this section, we reduce (19) and (21) to maximum fractional weighted matching problem. This reduction yields more efficient algorithms than those known for fractional/linear programming.

Note that, for a star $T(V \bigcup \{s\}, E)$ with center s, each path $\rho(i,j)$ consists of two edges (i,s) and (s,j), and the polytope defined in (19) is

$$\begin{aligned} \forall (i,j) \epsilon P: \quad & x_{i,j} \geq 0, \\ \forall i \epsilon V: \quad & \sum_{i<j} x_{i,j} + \sum_{j<i} x_{i,j} \leq 1, \\ & \max \left\{ \sum_{(i,j)\epsilon P} C_{i,j} \cdot x_{i,j} \right\}. \end{aligned} \tag{22}$$

Note that the matrix of linear program (22) is the incidence matrix of a complete graph with k vertices. Therefore, (22) is the problem of maximum fractional weighted matching (Lovasz and Plummer [46]).

Maximum fractional weighted matching problems can be easily reduced to classical maximum weighted matching problems in bipartite graphs. Let V' and V'' be two copies of the set V. Consider the bipartite graph $H = (V' \bigcup V'', U)$ with $k(k-1)$ edges. (We join i from the part V' with j from the part V'' by the edge (i,j) if $i \neq j$.) The matching polytope for H is described by the linear programming problem

$$\begin{aligned} \forall (i,j) \epsilon U: \quad & y_{i,j} \geq 0, \\ \forall i \epsilon V: \quad & \sum_{(i,j)\epsilon U} y_{i,j} \leq 1, \\ \forall j \epsilon V'': \quad & \sum_{(i,j)\epsilon U} y_{i,j} \leq 1, \\ & \max \left\{ \sum_{(i,j)\epsilon U} C_{i,j} \cdot y_{i,j} \right\}. \end{aligned} \tag{23}$$

Observe that each solution of (23) generates a solution of (19) by setting

$$x_{i,j} = \frac{y_{i,j} + y_{j,i}}{2}. \tag{24}$$

On the other hand, each solution of (19) generates a solution of (23) by $y_{i,j} = y_{j,i} = x_{i,j}$ for $1 \leq i < j \leq k$. The matrix of the linear programming problem (23) for a bipartite graph is totally unimodular; therefore, (23) has an integral solution for an arbitrary objective function (Lovasz and Plummer [46]). Due to (24), the linear programming problem (19) has a half-integer optimal solution for an arbitrary objective function. Reduction of the linear programming problem (21) to a maximum weighted matching problem can be obtained in a similar way by noting that there exists an optimal solution of (21) with x_q equal either $1/2$ or 1. (All vertices of fractional matching polytope are integer or half-integer; see Lovasz and Plummer [46].)

10. Discussion. Although the multiple alignment problem has frequently been studied, the first "performance guarantees" algorithm with $c = 2 - 2/k$ appeared in Gusfield [29]. We present a "performance guarantees" algorithm with $c = 2 - 3/k$ and conjecture that, for an arbitrary l, there exist polynomial approximation algorithms for multiple alignment with $c = 2 - l/k$. The merits and demerits of "performance guarantees" algorithms in comparison with former approaches are still unclear, but the empirical results of Gusfield [29] sound promising. It is worth emphasizing that

the results of the present paper on SP-alignment are mainly theoretical and the 3-star algorithm does not produce "good" alignments for sequences that are extremely different. (Of course, it can be argued that alignments of extremely different sequences are not usually of biological interest.) On the other hand, the combinatorial approach for deriving dynamic error bounds for star alignment should be incorporated in multiple alignment software. Lipman, Altschul, and Kececioglu [43] reported significant reduction of the computational volume of alignment graphs due to bounding procedures. Implementation of dynamic bounds for star alignment implies an increase in memory requirements (Spouge [60]) but allows fast and significant reduction in computational volume because of fast matching algorithms with $O(k^3)$ running time.

Acknowledgments. The author thanks M. Waterman for helpful discussions, as well as A. Kelmans and L. Khachian for useful comments on fractional programming. The author also thanks the referees for reading the manuscript carefully and providing many helpful suggestions.

REFERENCES

[1] G. M. Adel'son-Velsky, E. A. Dinic, and A. V. Karzanov, *Flow algorithms*, Nauka, Moscow, 1975. (In Russian.)

[2] R. K. Ahuja and V. V. S. Murty, *New lower planes for the network design problem*, Networks, 17 (1987), pp. 113–120.

[3] ———, *Exact and heuristic algorithms for the optimum communication spanning tree problem*, Transportation Sci., 21 (1987), pp. 163–170.

[4] S. F. Altschul, *Gap costs for multiple sequence alignment*, J. Theoret. Biol., 138 (1989), pp. 297–309.

[5] ———, *Amino acid substitution matrices from an information theoretic perspective.* J. Molecular Biol., 219 (1991), pp. 555–565.

[6] S. F. Altschul and D. J. Lipman, *Trees, stars, and multiple biological sequence alignment*, SIAM J. Appl. Math., 49 (1989), pp. 197–209.

[7] P. Argos, M. Vingron, and G. Vogt, *Protein sequence comparison: Methods and significance*, Protein Engineering, 4 (1991), pp. 375–383.

[8] M. L. Balinski, *Integer programming: Methods, uses and computation*, Management Sci., 12 (1965), pp. 253–313.

[9] D. J. Bacon and W. F. Anderson, *Multiple sequence alignment*, J. Molecular Biol., 191 (1986), pp. 153–161.

[10] W. Bains, MULTAN: *A program to align multiple* DNA *sequences*, Nucl. Acids Res., 14 (1986), pp. 159–177.

[11] G. J. Barton and M. J. E. Sternberg, *A strategy for multiple alignment of protein sequences*, J. Molecular Biol., 198 (1987), pp. 327–337.

[12] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis, *Linear approximation of shortest superstrings*, in Proc. 23rd ACM Sympos. Theory Comput., New Orleans, LA, May 6–8, 1991, pp. 328–336.

[13] H. Carrillo and D. Lipman, *The multiple sequence alignment problem in biology*, SIAM J. Appl. Math., 48 (1988), pp. 1073–1082.

[14] S. C. Chan, A. K. C. Wong, and D. K. Y. Chiu, *A survey of multiple sequence comparison methods*, Bull. Math. Biol., 54 (1992), pp. 563–598.

[15] C. Chappey, A. Danckaert, P. Dessen, and S. Hazout, MASH: *An interactive program for multiple alignment and consensus sequence construction for biological sequences*, Comput. Appl. Biosci., 7 (1991), pp. 195–202.

[16] A. Charnes and W. W. Cooper, *Programming with linear fractional functionals*, Naval Res. Logist. Quart., 9 (1962), pp. 181–186.

[17] B. V. Cherkassky, *The solution of a multicommodity flow problem. Mathematical methods in economy*, 13 (1977), pp. 143–151. (In Russian.)

[18] F. Corpet, *Multiple sequence alignment with hierarchical clustering*, Nucl. Acids Res., 16 (1988), pp. 10881–10890.

[19] D. Eppstein, Z. Galil, R. Giancarlo, and G. Italiano, *Efficient algorithms for sequence analysis*, in Proc. Second Workshop on Sequences: Combinatorics, Compression, Security, Transmission, Positano, Italy, June 17–21, 1991, to appear.

[20] D. FENG AND R. DOOLITTLE, *Progressive sequence alignment as a prerequisite to correct phylogenetic trees*, J. Molecular Evol., 25 (1987), pp. 351–360.
[21] D. E. FOULSER AND N. G. CORE, *Parallel computation of multiple biological sequence comparisons*, Comput. Biomed. Res., 23 (1990), pp. 310–331.
[22] M. L. FREDMAN, *Algorithms for computing evolutionary similarity measures with length independent gap penalties*, Bull. Math. Biol., 46 (1984), pp. 553–566.
[23] H. H. GABOW, *An efficient implementation of Edmonds' algorithm for maximum matching on graphs*, J. Assoc. Comput. Mach., 23 (1976), pp. 221–234.
[24] Z. GALIL, *Sequential and parallel algorithms for finding maximal matching in graphs*, Ann. Rev. Comput. Sci., 1 (1986), pp. 197–224.
[25] J. K. GALLANT, D. MAIER, AND J. A. STORER, *On finding minimal length superstrings*, J. Computers Syst. Sci., 20 (1980), pp. 50–58.
[26] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.
[27] O. GOTOH, *Alignment of three biological sequences with an efficient traceback procedure*, J. Theoret. Biol., 121 (1986), pp. 327–337.
[28] ———, *Consistency of optimal sequence alignments*, Bull. Math. Biol., 52 (1990), pp. 509–525.
[29] D. GUSFIELD, *Efficient method for multiple sequence alignment with guaranteed error bounds*, Tech. Report, Computer Science Division, University of California, Davis, CA, CSE-91-4, 1991; Bull. Math. Biol., 54 (1992), to appear.
[30] J. HEIN, *A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given*, Molecular Biol. Evol., 6 (1989), pp. 649–668.
[31] D. G. HIGGINS AND P. M. SHARP, CLUSTAL: *A package for performing multiple sequence alignment on a microcomputer*, Gene, 73 (1988), pp. 237–244.
[32] P. HOGEWEG AND B. HESPER, *The alignment of sets of sequences and the construction of phylogenetic trees, an integrated method*, J. Molecular Evol., 20 (1984), pp. 175–186.
[33] T. C. HU, *Optimum communication spanning trees*, SIAM J. Comput., 3 (1974), pp. 188–195.
[34] D. S. JOHNSON, J. K. LENSTRA, AND A. H. G. RINNOOY KAN, *The complexity of network design problem*, Network, 8 (1978), pp. 279–285.
[35] M. S. JOHNSON AND R. F. DOOLITTLE, *A method for simultaneous alignment of three or more amino acid sequences*, J. Molecular Evol., 23 (1986), pp. 267–278.
[36] S. KARLIN, M. MORRIS, G. GHANDOUR, AND M. Y. LEUNG, *Efficient algorithms for molecular sequence analysis*, Proc. Nat. Acad. Sci. U.S.A., 85 (1988), pp. 841–845.
[37] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[38] L. G. KHACHIAN, *A polynomial algorithm for linear programming*, Soviet Math. Dokl., 20 (1979), pp. 191–194.
[39] C. E. LAWRENCE AND A. A. REILLY, *An expectation maximization* (EM) *algorithm for the identification and characterization of common sites in unaligned biopolymer sequences*, Proteins, 7 (1990), pp. 41–51.
[40] A. M. LESK, M. LEVITT, AND C. CHOTHIA, *Alignment of the amino acid sequences of distantly related proteins using variable gap penalties*, Protein Engrg., 1 (1986), pp. 77–78.
[41] M. Y. LEUNG, B. E. BLAISDELL, C. BURGE, AND S. KARLIN, *An efficient algorithm for identifying matches with errors in multiple long molecular sequences*, J. Molecular Biol., 221 (1991), pp. 1367–1378.
[42] M. LI, *Towards a* DNA *sequencing theory*, in Proc. 31st IEEE Sympos. Foundations of Computer Science, 1990, St. Louis, MO, pp. 125–134.
[43] D. J. LIPMAN, S. F. ALTSCHUL, AND J. D. KECECIOGLU, *A tool for multiple sequence alignment*, Proc. Nat. Acad. Sci. U.S.A., 86 (1989), pp. 4412–4415.
[44] M. V. LOMONOSOV, *Combinatorial approaches to multiflow problems*, Discrete Appl. Math., 11 (1985), pp. 1–93.
[45] L. LOVASZ, *2-matchings and 2-covers of hypergraphs*, Acta Math. Acad. Sci. Hungar., 26 (1975), pp. 433–444.
[46] L. LOVASZ AND M. D. PLUMMER, *Matching Theory*, Elsevier, Amsterdam, the Netherlands, 1986.
[47] D. MAIER, *The complexity of some problems on subsequences and supersequences*, J. Assoc. Comput. Mach., 25 (1978), pp. 322–336.
[48] H. MARTINEZ, *Flexible multiple sequence alignment program*, Nucl. Acids Res., 16 (1988), pp. 1683–1691.
[49] M. MURATA, J. S. RICHARDSON, AND J. L. SUSSMAN, *Simultaneous comparison of three protein sequences*, Proc. Nat. Acad. Sci. U.S.A., 82 (1985), pp. 3073–3077.
[50] L. PATTHY, *Detecting homology of distantly related proteins with consensus sequences*, J.

Molecular Biol., 198 (1987), pp. 567–577.
[51] J. POSFAI, A. S. BHAGWAT, G. POSFAI, AND R. J. ROBERTS, *Predictive motifs derived from cytosine methyltransferases*, Nucl. Acids Res., 17 (1989), pp. 2421–2435.
[52] C. QUEEN, M. N. WEGMAN, AND L. J. KORN, *Improvements to a program for* DNA *analysis: A procedure to find homologies among many sequences*, Nucl. Acids Res., 10 (1982), pp. 449–457.
[53] M. A. ROYTBERG, *A search for common patterns of many sequences*, Comput. Appl. Biol. Sci., 8 (1992), pp. 57–64.
[54] D. SANKOFF, *Minimum mutation tree of sequences*, SIAM J. Appl. Math., 28 (1975), pp. 35–42.
[55] ———, *Simultaneous solution of the* RNA *folding, alignment, and protosequence problems*, SIAM J. Appl. Math., 45 (1985), pp. 810–825.
[56] M. SANTIBANEZ AND K. ROHDE, *A multiple alignment program for protein sequences*, Comput. Appl. Biosci., 3 (1987), pp. 111–114.
[57] G. D. SCHULER, S. F. ALTSCHUL, AND D. J. LIPMAN, *A workbench for multiple alignment construction and analysis*, PROTEINS: Structure, Functions, and Genetics, 9 (1991), pp. 180–190.
[58] H. O. SMITH, T. M. ANNAU, AND S. CHANDRASEGARAN, *Finding sequence motifs in groups of functionally related proteins*, Proc. Nat. Acad. Sci. U.S.A., 87 (1990), pp. 826–830.
[59] E. SOBEL AND H. MARTINEZ, *A multiple sequence alignment program*, Nucl. Acids Res., 16 (1986), pp. 363–374.
[60] J. L. SPOUGE, *Speeding up dynamic programming algorithms for finding optimal lattice paths*, SIAM J. Appl. Math., 49 (1989), pp. 1552–1566.
[61] G. D. STORMO AND G. W. HARTZELL III, *Identifying protein-binding sites from unaligned* DNA *fragments*, Proc. Nat. Acad. Sci. U.S.A., 86 (1989), pp. 1183–1187.
[62] S. SUBBIAH AND S. C. HARRISON, *A method for multiple sequence alignment with gaps*, J. Molecular Biol., 209 (1989), pp. 539–548.
[63] K. TAJIMA, *Multiple* DNA *and protein sequence alignment on a workstation and a supercomputer*, Comput. Appl. Biosci., 4 (1988), pp. 467–471.
[64] J. TARHIO AND E. UKKONEN, *A greedy approximation algorithm for constructing shortest common superstrings*, Theoret. Comput. Sci., 57 (1988), pp. 131–145.
[65] W. R. TAYLOR, *The classification of amino acid conservation*, J. Theoret. Biol., 119 (1986), pp. 205–218.
[66] ———, *Multiple sequence alignment by a pairwise algorithm*, Comput. Appl. Biosci., 3 (1987), pp. 81–87.
[67] V. G. TIMKOVSKY, *The complexity of subsequence, supersequences, and related problems*, Kibernetika, 5 (1989), pp. 1–13. (English translation.)
[68] J. S. TURNER, *Approximation algorithms for the shortest common superstring problem*, Inform. and Comput., 83 (1989), pp. 1–20.
[69] M. VIHINEN, *An algorithm for simultaneous comparison of several sequences*, Comput. Appl. Biosci., 4 (1988), pp. 89–92.
[70] M. VINGRON AND P. ARGOS, *A fast and sensitive multiple sequence alignment algorithm*, Comput. Appl. Biosci., 5 (1989), pp. 115–121.
[71] ———, *Motif recognition and alignment for many sequences by comparison of dot-matrices*, J. Molecular Biol., 218 (1991), pp. 33–43.
[72] M. S. WATERMAN, *Multiple sequence alignment by consensus*, Nucl. Acids Res., 16 (1986), pp. 9095–9102.
[73] ———, *Sequence alignments*, in Mathematical Analysis of DNA Sequences., M.S. Waterman, ed., CRC, Boca Raton, FL, 1989, pp. 53–92.
[74] M. S. WATERMAN, R. ARRATIA, AND D. J. GALAS, *Pattern recognition in several sequences: Consensus and alignment*, Bull. Math. Biol., 46 (1984), pp. 515–527.
[75] M. S. WATERMAN AND M. D. PERLWITZ, *Line geometries for sequence comparison*, Bull. Math. Biol., 46 (1984), pp. 567–577.
[76] M. S. WATERMAN, T. F. SMITH, AND W. A. BEYER, *Some biological sequence metrics*, Adv. Math., 20 (1976), pp. 367–387.

References

[6] **Trees, Stars, and Multiple Biological Sequence Alignment**
Stephen F. Altschul; David J. Lipman
SIAM Journal on Applied Mathematics, Vol. 49, No. 1. (Feb., 1989), pp. 197-209.
Stable URL:
http://links.jstor.org/sici?sici=0036-1399%28198902%2949%3A1%3C197%3ATSAMBS%3E2.0.CO%3B2-0

[13] **The Multiple Sequence Alignment Problem in Biology**
Humberto Carrillo; David Lipman
SIAM Journal on Applied Mathematics, Vol. 48, No. 5. (Oct., 1988), pp. 1073-1082.
Stable URL:
http://links.jstor.org/sici?sici=0036-1399%28198810%2948%3A5%3C1073%3ATMSAPI%3E2.0.CO%3B2-O

[54] **Minimal Mutation Trees of Sequences**
David Sankoff
SIAM Journal on Applied Mathematics, Vol. 28, No. 1. (Jan., 1975), pp. 35-42.
Stable URL:
http://links.jstor.org/sici?sici=0036-1399%28197501%2928%3A1%3C35%3AMMTOS%3E2.0.CO%3B2-2

[55] **Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems**
David Sankoff
SIAM Journal on Applied Mathematics, Vol. 45, No. 5. (Oct., 1985), pp. 810-825.
Stable URL:
http://links.jstor.org/sici?sici=0036-1399%28198510%2945%3A5%3C810%3ASSOTRF%3E2.0.CO%3B2-U

LINKED CITATIONS

[60] **Speeding Up Dynamic Programming Algorithms for Finding Optimal Lattice Paths**
John L. Spouge
SIAM Journal on Applied Mathematics, Vol. 49, No. 5. (Oct., 1989), pp. 1552-1566.
Stable URL:
http://links.jstor.org/sici?sici=0036-1399%28198910%2949%3A5%3C1552%3ASUDPAF%3E2.0.CO%3B2-U