Bulletin of Mathematical Biology Vol. 46, No. 4, pp. 567–577, 1984.
Printed in Great Britain

0092-8240/84$3.00 + 0.00
Pergamon Press Ltd.

LINE GEOMETRIES FOR SEQUENCE COMPARISONS*

■ MICHAEL S. WATERMAN and MARCELA D. PERLWITZ
Department of Mathematics and of Molecular Biology,
Los Angeles, CA 90089-1113, U.S.A.

Well-known dynamic programming algorithms exist for comparing two finite sequences in $O(N^2)$ time and storage, where N is the common sequence length. Extensions to the comparison of M finite sequences require $O((2N)^M)$ time and storage, making such algorithms difficult even for $M = 3$. A simple generalization of the sequences makes it possible to obtain some results about the geometry of sequence alignments. These ideas suggest heuristic approaches to problems of comparing several sequences. If M sequences are known to be related by a binary tree, they can be aligned in $O(MN^2)$ time and $O(N^2 + NM)$ storage.

1. Introduction. Mathematical methods for comparing two finite genetic sequences began with Ulam (1972), Sankoff (1972) and Sellers (1974). A corresponding development exists in computer science, starting with Wagner and Fischer (1974). Advances continue to be made for these algorithms and they are routinely applied by scientists studying genetic sequences such as DNA. Sankoff and Kruskal (1983) have edited a book which surveys the application of dynamic programming to these and other areas of science.

Unfortunately, there is a lack of practical algorithms for comparison of three or more sequences. Sankoff (1975) gives an algorithm which, given a finite tree T with terminal vertices identified as sequences, finds sequences for the interior vertices which minimize the sum of the lengths of edges in T. If there are M sequences of length N, his algorithm uses $(2N)^M L$ time and $O(N^M)$ storage, where L is the number of interior vertices. Independently, Waterman *et al.* (1976) give a related algorithm. Choosing Sankoff's tree T to connect all M sequences to a single point and the ρ of Waterman *et al.* to be

$$\rho(x_1, x_2, \ldots, x_m) = \min_{\lambda} \sum_{i=1}^{m} d(x_i, \lambda),$$

where $d(\cdot\,,\cdot)$ is the metric on the alphabet, then the method is that of Sankoff. The metric ρ is related to parsimony (Fitch, 1971). For $d(x, y) = 1$ if $x \neq y$, the above ρ gives

* This work was supported by a grant from the System Development Foundation.

$$\rho(\mathrm{A, A, A, T, A, A, A}) = \sum_{i=1}^{7} d(x_i, \mathrm{A}) = 1,$$

and

$$\rho(\mathrm{C, G, C, G, C, G, C}) = \sum_{i=1}^{7} d(x_i, \mathrm{C}) = 3.$$

In this paper we explore the implications of an idea distinct from parsimony. Essentially, the weighted average of two sequences is defined. In speech recognition work, there are definitions of average trajectories and of average sequences (Rabiner and Wilpon, 1979, 1980; Kruskal and Liberman, 1983), but no results or methods related to those we obtain are given by these authors. We present some observations about the geometry of such sequence comparisons. We refer to these geometries as line geometries because any two points (sequences) can be joined by a straight line in the metric space. This geometry has some highly non-Euclidean properties and is not currently well understood. Busemann (1955) studies the geometry of geodesics and refers to spaces such as we study as "straight". In the final section we discuss the problems of aligning several sequences with these techniques. A useful application is a method for aligning two sets of sequences, each set of which has already been aligned. While there does not seem to be much hope for M sequences of unknown relationship, if the M sequences are related by a binary tree they can be aligned in $O(MN^2)$ steps by a heuristic method naturally suggested by the geometry.

2. Weighted Average Sequences and Their Geometry. For our purposes, a new but simple concept of sequence is required along with a specific family of metrics on the letters of the sequence. First, if the original sequences are finite words over an alphabet A, define a weighted average sequence to be a finite sequence $\mathbf{a} = a_1, a_2 \ldots a_n$ where each a_i has the form $a_i = (p_0, p_1, p_2, \ldots)$ where $p_i \geqslant 0$ and

$$\sum_{i \geqslant 0} p_i = 1.$$

If p_i corresponds to the proportion of the ith element of A and p_0 to the proportion of Δ, it is then easy to convert a usual sequence into a weighted average sequence. The letter Δ is thought of as a space, indicating a deletion in the sequence in which it appears or an insertion in the opposite sequence.

To compare two letters $a = (p_0, p_1, \ldots)$ and $b = (q_0, q_1, \ldots)$, simply set

$$d(a, b) = \left(\sum_{i \geq 0} w_i \mid p_i - q_i \mid^{\alpha} \right)^{1/\alpha}$$

where w_i are weighting factors and $\alpha \geq 1$ is a constant. It is well known that d is a metric on our set of letters.

To compute the distance $D(\mathbf{a}, \mathbf{b})$ between two weighted average sequences, the usual dynamic programming algorithm is employed. Here $\mathbf{a} = a_1 a_2 \ldots a_n$ and $\mathbf{b} = b_1 b_2 \ldots b_m$. If $D_{ij} = D(a_1 \ldots a_i, b_1 \ldots b_j)$, $D_{0j} = D(\underset{\sim}{\Delta}, b_1 \ldots b_j)$, $D_{i0} = D(a_1 \ldots a_i, \underset{\sim}{\Delta})$, $D_{00} = 0$, then

$$D_{i,j} = \min\{D_{i-1,j} + d(a_i, \Delta), D_{i-1,j-1} + d(a_i, b_j), D_{i,j-1} + d(\Delta, b_j)\}.$$

Throughout $\Delta = (1, 0, \ldots)$ when used as a letter and $\underset{\sim}{\Delta} = \Delta\Delta \ldots$ when used as a sequence. Of course $D_{n,m} = D(\mathbf{a}, \mathbf{b})$.

Corresponding to $\mathbf{a}$, $\mathbf{b}$ and $D(\mathbf{a}, \mathbf{b})$ is a set of optimal alignments of $\mathbf{a}$ with $\mathbf{b}$. An alignment is a row listing of $\mathbf{a} = a_1 a_2 \ldots a_n$ where Δs can be inserted among the a_is under which $\mathbf{b} = b_1 b_2 \ldots b_m$ is written in a similar form. For example *aaca* can be aligned with *acc* by

$$\begin{matrix} a & a & c & a \\ a & \Delta & c & c. \end{matrix}$$

The score of an alignment is the sum of the pairwise distances of the "matching" letters. An optimal alignment is one whose score is $D(\mathbf{a}, \mathbf{b})$. If the length of such an alignment is L, we write

$$\begin{matrix} a_1^* a_2^* \ldots a_L^* \\ b_1^* b_2^* \ldots b_L^* \end{matrix}$$

where the subsequence of $\mathbf{a}^*$ not equal to Δ is $\mathbf{a}$ and where the subsequence of $\mathbf{b}^*$ not equal to Δ is $\mathbf{b}$.

For an optimal alignment of $\mathbf{a}$ and $\mathbf{b}$ define $\mathbf{c}(\lambda) = \lambda\mathbf{a} \oplus (1 - \lambda)\mathbf{b}$ where $c_i(\lambda) = \lambda a_i^* + (1 - \lambda) b_i^*$ and the last "+" sign is a simple vector addition. In case $\lambda = \frac{1}{2}$, $\mathbf{c}(\frac{1}{2})$ is an equal weighting of a_i^* and b_i^* from the optimal alignment of $\mathbf{a}$ and $\mathbf{b}$. One might suspect that $\mathbf{c}(\frac{1}{2})$ is midway between $\mathbf{a}$ and $\mathbf{b}$. More than that turns out to be true and Theorem 1 states that the metric space is a line geometry.

THEOREM 1. Let

$$\mathbf{c}(\lambda) = \lambda\mathbf{a} \oplus (1 - \lambda)\mathbf{b}.$$

Then

$$D(\mathbf{a}, \mathbf{b}) = D[\mathbf{a}, \mathbf{c}(\lambda)] + D[\mathbf{b}, \mathbf{c}(\lambda)]$$

and

$$D[\mathbf{a}, \mathbf{c}(\lambda)] = (1-\lambda)D(\mathbf{a}, \mathbf{b}).$$

Proof

$$D[\mathbf{a}, \mathbf{c}(\lambda)] \leqslant \sum_{i=1}^{L} d[a_i^*, c_i(\lambda)]$$

$$= \sum_{i=1}^{L} [\Sigma_j w_j |p_j - [\lambda p_j + (1-\lambda)q_j]|^\alpha]^{1/\alpha}$$

$$= (1-\lambda) \sum_{i=1}^{L} d(a_i^*, b_i^*) = (1-\lambda)D(\mathbf{a}, \mathbf{b}).$$

In the same manner, $D[\mathbf{c}(\lambda), \mathbf{b}] \leqslant \lambda D(\mathbf{a}, \mathbf{b})$ and $D[\mathbf{a}, \mathbf{c}(\lambda)] + D[\mathbf{c}(\lambda), \mathbf{b}] \leqslant D(\mathbf{a}, \mathbf{b})$. The triangle inequality implies each of the inequalities are equalities.

COROLLARY

$$D[\mathbf{c}(\lambda_1), \mathbf{c}(\lambda_2)] = |\lambda_1 - \lambda_2| D(\mathbf{a}, \mathbf{b}).$$

Proof. We cannot assume $\mathbf{c}(\lambda_1)$ and $\mathbf{c}(\lambda_2)$ are the result of the same alignment. Still

$$D(\mathbf{a}, \mathbf{b}) = D(\mathbf{a}, \mathbf{c}(\lambda_1)) + D(\mathbf{c}(\lambda_1), \mathbf{c}(\lambda_2)) + D(\mathbf{c}(\lambda_2), \mathbf{b})$$

and the corollary follows.

The theorem implies that a weighted average sequence can be found to represent any point on the line between two sequences. While the converse of the theorem is not true, it has a coordinate by coordinate version.

THEOREM 2. If $\mathbf{c}$ satisfies $D(\mathbf{a}, \mathbf{c}) + D(\mathbf{c}, \mathbf{b}) = D(\mathbf{a}, \mathbf{b})$, then each $c_i = \lambda_i a_i^* + (1-\lambda_i) b_i^*$ for some optimal alignment of $\mathbf{a}$ and $\mathbf{b}$.

Proof. By inserting $\underset{\Delta}{\Delta}$ into optimal $\mathbf{a}, \mathbf{c}$ and $\mathbf{c}, \mathbf{b}$ alignments, the alignments can be assumed to be of equal length:

$$\begin{matrix} a_1^* \ldots a_L^* \\ c_1^* \ldots c_L^* \end{matrix},$$

$$\begin{matrix} c_1^* \ldots c_L^* \\ b_1^* \ldots b_L^* \end{matrix}$$

Because $D(\mathbf{a}, \mathbf{b}) = D(\mathbf{a}, \mathbf{c}) + D(\mathbf{c}, \mathbf{b})$, the implied $\mathbf{a}, \mathbf{b}$ alignment is optimal. Moreover $d(a_i^*, b_i^*) = d(a_i^*, c_i^*) + d(c_i^*, b_i^*)$ and the result follows.

At this point it might be conjectured that the geometry for more than two sequences is approximately Euclidean. If this were true, then efficient algorithms for comparing sequences are immediately suggested. Unfortunately the geometrical properties of even three sequences is far from simple. Let $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$, be given sequences and define $\mathbf{b}(\lambda) = \lambda\mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_2$ and $\mathbf{c}(\lambda) = \lambda\mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_3$ for $\lambda \in [0, 1]$. Now

$$D(\mathbf{b}(1), \mathbf{c}(1)) = 0$$

and

$$D(\mathbf{b}(0), \mathbf{c}(0)) = D(\mathbf{a}_2, \mathbf{a}_3)$$

and, if $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ formed a triangle in plane geometry, $D(\mathbf{b}(\lambda), \mathbf{c}(\lambda)) = (1-\lambda)D(\mathbf{a}_2, \mathbf{a}_3)$ would hold. This equation need only hold at $\lambda = 0, 1$. For an example, we take three portions of sequences from 16*S* rRNA sequences of *B. stearothermophilus* ($\mathbf{a}_1$), *D. discoideum* ($\mathbf{a}_2$) and *E. coli* ($\mathbf{a}_3$) (Woese *et al.*, 1983; Table 31). In Fig. 1, values of $D(\mathbf{b}(\lambda), \mathbf{c}(\lambda))$ are plotted in order to show the deviation from Euclidean geometry.

If all sequences are of equal length and the deletion weight is large, then the *i*th column in any alignment is composed of the *i*th members of the original sequences. In this extreme case the resulting line geometry is Euclidean.

3. Algorithms. We now turn to consideration of algorithms for M sequences where $M \geqslant 3$. These ideas do not seem to suggest practical methods for aligning M sequences of unknown relationship However the problem of aligning M sequences, when a binary phylogenetic tree is assumed, does have a practical heuristic solution. We turn first to a simple but important problem.

3.1 Aligning alignments. Suppose two sets of sequences $\mathbf{a}_1, \mathbf{a}_2 \ldots \mathbf{a}_k$ and $\mathbf{b}_1, \mathbf{b}_2 \ldots \mathbf{b}_l$ have been aligned by some method. Each such alignment can be easily made into a weighted average sequence $\mathbf{a}_*$ and $\mathbf{b}_*$. The metric, $D(\cdot,\cdot)$, above can be applied to align these alignments. Notice that $\lambda\mathbf{a}_* \oplus (1-\lambda)\mathbf{b}_*$ can be formed from any alignment which gives $D(\mathbf{a}_*, \mathbf{b}_*)$ but that the number of sequences involved, k and l, do not enter into computing $D(\mathbf{a}_*, \mathbf{b}_*)$.

3.2 Center of gravity sequences. Consider three sequences $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$. $\mathbf{e}_2 = \frac{1}{2}\mathbf{a}_1 \oplus \frac{1}{2}\mathbf{a}_2$ occupies the midpoint of a line between $\mathbf{a}_1$ and $\mathbf{a}_2$. If all distances had the properties of Euclidean geometry, the center of gravity is a point on a line from the midpoint $\mathbf{e}_2$ to $\mathbf{a}_3$, two-thirds of the length from $\mathbf{a}_3$ and one-third from $\mathbf{e}_2$. Therefore the desired sequence is $\mathbf{e}_3 = \frac{1}{3}\mathbf{a}_3 \oplus \frac{2}{3}[\mathbf{e}_2]$.

The algorithm of the previous paragraph generalizes to M sequences, $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M$ by

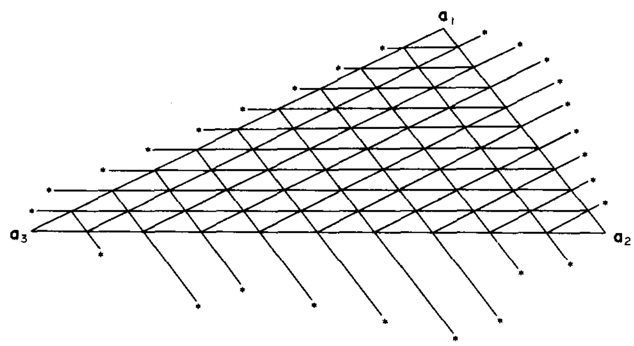


Figure 1. For the three sequences below, the distances $D(\lambda\mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_2, \lambda\mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_3)$, $D(\lambda\mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_3, \lambda\mathbf{a}_2 \oplus (1-\lambda)\mathbf{a}_3)$ and $D(\lambda\mathbf{a}_2 \oplus (1-\lambda)\mathbf{a}_1, \lambda\mathbf{a}_2 \oplus (1-\lambda)\mathbf{a}_3$ are plotted for

$$\lambda = \frac{1}{10}, \ldots, \frac{9}{10}.$$

Such distances are represented by the length of lines joining the appropriate points of the sides of the triangle ending in *. If the line geometry were Euclidean none of these lines would extend beyond the triangle.

Sequence $\mathbf{a}_1$: is from *B. stearothermophilus*:

CAACCCUCGCCUCUAGUCACUCUAGAGGGGAAGGUGGGGA

Sequence $\mathbf{a}_2$: is from *D. discoideum*:

AGACCUCGACCUGCUAACCUUCUUAGAGGGGAAGUCCGAGG

Sequence $\mathbf{a}_3$: is from *E. coli*:

CCACCCUUAUCCUUUGUAACUCAAAGGAGGAAGGUGGGGA

$$d(p, q) = 4|p_o - q_o| + \sum_{i=1}^{4} |p_i - q_i|.$$

$$\mathbf{e}_1 = \mathbf{a}_1,$$

$$\mathbf{e}_2 = \frac{1}{2}\mathbf{a}_2 \oplus \frac{1}{2}\mathbf{e}_1,$$

$$\mathbf{e}_3 = \frac{1}{3}\mathbf{a}_3 \oplus \frac{2}{3}\mathbf{e}_2,$$

$$\ldots$$

$$\mathbf{e}_M = \frac{1}{M}\mathbf{a}_M \oplus \frac{M-1}{M}\mathbf{e}_{M-1}.$$

This algorithm runs in time $(M-1)O(N^2)$ where $O(N^2)$ is the time required to align two sequences of length N. The storage required is dominated by $O(N^2)$, that required to align two sequences. The alignment of M sequences uses MN storage.

In the above discussion, the M sequences were assumed to be equally weighted. Unequal weighting of sequences is easily included.

We can locate all sequences $\lambda \mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_2$ on a line. The main difficulty with the proposed algorithm is that, as illustrated in Section 2

$$D[\lambda \mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_2, \lambda \mathbf{a}_1 \oplus (1-\lambda)\mathbf{a}_3]$$

cannot be assumed to be $(1-\lambda)D(\mathbf{a}_2, \mathbf{a}_3)$. This implies that

$$\frac{1}{3}\mathbf{a}_3 \oplus \frac{2}{3}\left(\frac{1}{2}\mathbf{a}_2 \oplus \frac{1}{2}\mathbf{a}_1\right),$$

$$\frac{1}{3}\mathbf{a}_2 \oplus \frac{2}{3}\left(\frac{1}{2}\mathbf{a}_3 \oplus \frac{1}{2}\mathbf{a}_1\right),$$

and

$$\frac{1}{3}\mathbf{a}_1 \oplus \frac{2}{3}\left(\frac{1}{2}\mathbf{a}_2 \oplus \frac{1}{2}\mathbf{a}_3\right)$$

might all have different metric properties. Set

$$\mathbf{b}_1 = \frac{1}{2}\mathbf{a}_1 \oplus \frac{1}{2}\mathbf{a}_2,$$

$$\mathbf{b}_2 = \frac{1}{2}\mathbf{a}_1 \oplus \frac{1}{2}\mathbf{a}_3,$$

$$\mathbf{b}_3 = \frac{1}{2}\mathbf{a}_2 \oplus \frac{1}{2}\mathbf{a}_3.$$

Notice that, by Theorem 1 and the triangle inequality,

$$D(\mathbf{a}_1, \mathbf{a}_2) + D(\mathbf{a}_2, \mathbf{a}_3) + D(\mathbf{a}_3, \mathbf{a}_1) \geqslant D(\mathbf{b}_1, \mathbf{b}_2) + D(\mathbf{b}_2, \mathbf{b}_3) + D(\mathbf{b}_3, \mathbf{b}_1)$$

and, if equality holds, $\mathbf{a}_1 = \mathbf{a}_2 = \mathbf{a}_3$. We find the algorithm

$$(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) \leftarrow \left(\frac{1}{2}\mathbf{a}_1 \oplus \frac{1}{2}\mathbf{a}_2, \frac{1}{2}\mathbf{a}_2 \oplus \frac{1}{2}\mathbf{a}_3, \frac{1}{2}\mathbf{a}_3 \oplus \frac{1}{2}\mathbf{a}_2\right) \tag{1}$$

Go to (1) if $D(\mathbf{a}_1, \mathbf{a}_2) + D(\mathbf{a}_2, \mathbf{a}_3) + D(\mathbf{a}_3, \mathbf{a}_1) > \epsilon$. (2)
Otherwise, stop.

converges very slowly. For an example of this iterated midpoint algorithm see Table 1 and Fig. 2. Of course, replacing $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ by the three possible center of gravity sequences should cause more rapid convergence but this does not seem to help much. We have not found that this algorithm converges exponentially.

TABLE I

Edge Lengths of Triangles obtained from the Iterated Midpoint Algorithm

Iteration No.	$D(\mathbf{a}_1, \mathbf{a}_2)$	$D(\mathbf{a}_1, \mathbf{a}_2)$	$D(\mathbf{a}_2, \mathbf{a}_3)$
0	12.000	33.000	35.000
1	13.000	17.500	18.500
2	8.000	12.750	13.750
3	8.250	9.125	9.375
4	6.494	7.685	8.311
5	6.627	6.776	6.713
6	5.852	6.163	6.601
7	5.951	5.983	5.878
8	5.302	5.614	5.946
9	5.607	5.596	5.463
10	5.305	5.331	5.590
11	5.360	5.352	5.242
12	5.137	5.172	5.351
13	5.211	5.166	5.085
14	4.992	5.054	5.180
15	5.065	5.032	4.979
16	4.897	4.944	5.023
17	4.933	4.922	4.889
18	4.790	4.860	4.906
19	4.834	4.814	4.801

The initial triangle has vertices $\mathbf{a}_1$ from *D. discodeum*, $\mathbf{a}_2$ from *S. cerevisiae* and $\mathbf{a}_3$ from *H. volcanii*. New triangles are formed by joining the midpoints of the previous triangle by the algorithm of Section 3.2. The edge lengths of these triangles are tabulated above.

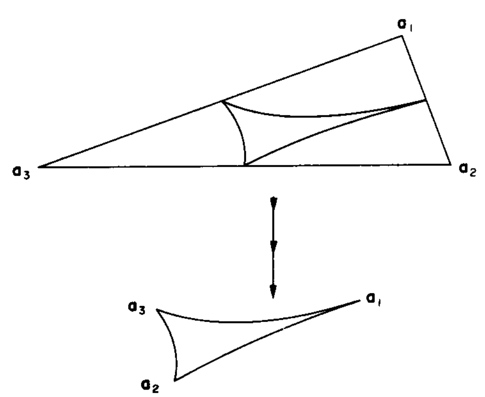


Figure 2. The initial triangle has vertices $\mathbf{a}_1$ from *D. discoideum*, $\mathbf{a}_2$ from *S. cerevisiae* and $\mathbf{a}_3$ from *H. volcanii*. Form a new triangle by joining the midpoints of previous triangle by the algorithms of Section 3.2.

3.3 Sequences related by a binary tree. As mentioned in the introduction, Sankoff's work assumes a binary tree T relating M sequences. If $M > 3$, we show that there is a natural algorithm which terminates in $O(MN^2)$ steps.

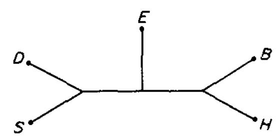
A binary tree with M external nodes has $M - 3$ interior branches. Combine the exterior nodes by $\oplus$ until only one node remains. There is no ambiguity about order once an interior branch is chosen to be the last remaining branch. Then these last two weighted average sequences are combined. The computation time is $O(MN^2)$.

An alignment of $M > 3$ sequences can be obtained by aligning each of the M original sequences with the final weighted average sequence. Above we did not specify the choice of weights. That is, $\lambda \mathbf{a}_2 \oplus (1 - \lambda)\mathbf{b}$ involves a choice of λ. Without additional information, it seems reasonable to weight the sequences proportional to their number as in Section 3.2. For example,

$$\frac{1}{3}\mathbf{c} \oplus \frac{2}{3}\left(\frac{1}{2}\mathbf{a} \oplus \frac{1}{2}\mathbf{b}\right)$$

gives each sequence equal weight. Other choices are clearly possible.

As an example of this procedure, consider $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_5$ as given in Fig. 3.



E. Coli
CAACCCUUAUCCUUUGUAACUCAAAGGAGGAAGGUGGGGA
B. stearo.
CAACCCUCGCCUCUAGUCACUCUAGAGGGGAAGGUGGGGA
H. volcanii
AGACCCGCACUUCUAAUUACAUUAGAAGGGAAGGAACGGG
D. discoideum
AGACCUCGACCUGCUAACCUUCUUAGAGGGGAAGUCCGAGG
S. cerevisiae
AGACCUUAACCUACUAAACUUCUUAGAGGGGAAGUUUGAGG

Figure 3. Assumed tree for the sequences. (From Woese *et al.*, 1983.)

The combination

$$\mathbf{b} = \frac{1}{5}\mathbf{a}_5 \oplus \frac{4}{5}\left[\frac{1}{2}\left(\frac{1}{2}\mathbf{a}_1 \oplus \frac{1}{2}\mathbf{a}_2\right) \oplus \frac{1}{2}\left(\frac{1}{2}\mathbf{a}_3 \oplus \frac{1}{2}\mathbf{a}_4\right)\right]$$

is obtained. By aligning each of $\mathbf{a}_1, \ldots, \mathbf{a}_5$ with $\mathbf{b}$,

$$d(a, b) = 4\left|p_0 - q_0\right| + \sum_{i=1}^{4}\left|p_i - q_i\right|,$$

the overall alignment is obtained. This alignment agrees with that given by Woese *et al.* (Table 31) who obtain it by their phylogenetic analysis of 16*S* like RNAs.

E. Coli
CAACCCUUAUCCΔUUUGUAACUCAAAGGAGGAAGGUGGGGA
B. stearo.
CAACCCUCGCCUΔCUAGUCACUCUAGAGGGGAAGGUGGGGA
H. volcanii
AGACCCGCACUUΔCUAAUUACAUUAGAAGGGAAGGAACGGG
D. discoideum
AGACCUCGACCUGCUAACCUUCUUAGAGGGGAAGUCCGAGG
S. cerevisiae
AGACCUUAACCUACUAAACUUCUUAGAGGGGAAGUUUGAGG

The authors are grateful to Eric Silber who wrote programs for the sequence comparison and alignment algorithms reported in this paper.

LITERATURE

Busemann, H. 1955. *The Geometry of the Geodesics*. Academic Press, New York.

Fitch, W. M. 1971. "Towards Defining the Course of Evolution: Minimum Change for a Specific Tree Topology." *Syst. Zool.* **20**, 406–416.

Kruskal, J. B. and M. Liberman. 1983. In *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*, Ed. D. Sankoff and J. B. Kruskal, pp. 125–160. Addison-Wesley, London.

Rabiner, L. R. and J. G. Wilpon. 1979. "Considerations in Applying Clustering Techniques to Speaker-independent Word Recognition." *J. Acoustic. Soc. Am.* **66**, 663–673.

—— and ——. 1980. "A Simplified, Robust Training Procedure for Speaker-trained, Isolated-word Recognition Systems." *J. Acoustic. Soc. Am.* **68**, 1271–1276.

Sankoff, D. 1972. "Matching Sequences under Deletion-Insertion Constraints." *Proc. natn. Acad. Sci. U.S.A.* **69**, 4–6.

——. 1975. "Minimal Mutation Trees of Sequences." *SIAM J. appl. Math.* **28**, 35–42.

—— and J. B. Kruskal. 1983. *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*. Addison-Wesley, London.

Sellers, P. H. 1974. "On the Theory and Computation of Evolutionary Distances." *SIAM J. appl. Math.* **26**, 787–793.

Ulam, S. M. 1972. In *Applications of Number Theory to Numerical Analysis*, Ed. S. K. Zaremba, pp. 1–10. Academic Press, New York.

Wagner, R. A. and M. J. Fischer. 1974. "The String-to-String Correction Problem." *J. Ass. comp. Mach.* **21**, 169–173.

Waterman, M. S., T. F. Smith and W. A. Beyer. 1976. "Some Biological Sequence Metrics." *Adv. Math.* **20**, 367–387.

Woese, C. R., R. Gutell, R. Gupta and H. F. Noller. 1983. "Detailed Analysis of the Higher-order Structure of 16*S*-like Ribosonal Ribonucleic Acids." *Microbiol. Rev.* **47**, 621–669.