JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 7, Numbers 1/2, 2000
Mary Ann Liebert, Inc.
Pp. 261–276

Evaluation Measures of Multiple Sequence Alignments

GASTON H. GONNET, CHANTAL KOROSTENSKY, and STEVE BENNER

ABSTRACT

Multiple sequence alignments (MSAs) are frequently used in the study of families of protein sequences or DNA/RNA sequences. They are a fundamental tool for the understanding of the structure, functionality and, ultimately, the evolution of proteins. A new algorithm, the Circular Sum (CS) method, is presented for formally evaluating the quality of an MSA. It is based on the use of a solution to the Traveling Salesman Problem, which identifies a circular tour through an evolutionary tree connecting the sequences in a protein family. With this approach, the calculation of an evolutionary tree and the errors that it would introduce can be avoided altogether. The algorithm gives an upper bound, the best score that can possibly be achieved by any MSA for a given set of protein sequences. Alternatively, if presented with a specific MSA, the algorithm provides a formal score for the MSA, which serves as an absolute measure of the quality of the MSA. The CS measure yields a direct connection between an MSA and the associated evolutionary tree. The measure can be used as a tool for evaluating different methods for producing MSAs. A brief example of the last application is provided. Because it weights all evolutionary events on a tree identically, but does not require the reconstruction of a tree, the CS algorithm has advantages over the frequently used sum-of-pairs measures for scoring MSAs, which weight some evolutionary events more strongly than others. Compared to other weighted sum-of-pairs measures, it has the advantage that no evolutionary tree must be constructed, because we can find a circular tour without knowing the tree.

Key words: multiple sequence alignment, phylogenetic tree, scoring function, TSP, evolution.

1. INTRODUCTION

One of the most important applications of the data being generated from genome sequencing projects is to reconstruct the evolutionary histories of proteins, nucleic acids, and the organisms that carry them. A model for an evolutionary history typically consists of three parts: (a) an evolutionary tree, which shows the relationships of evolutionary objects (proteins, for example), (b) a multiple sequence alignment (MSA), which shows the evolutionary relationships between parts of these objects (in this case, individual amino acids in the protein sequence), and (c) reconstructed ancestral sequences, models for objects that were intermediates in the evolution of the family. Evolutionary histories are important in deducing biological function in biomolecules, predicting the folded conformation of protein sequences, and reconstructing the history of life on Earth.

Institute for Scientific Computing, ETH Zurich, 8092 Zurich, Switzerland.

Example. In the MSA in Figure 1, five sequences are aligned ($n = 5$). In this example, we assume that the sequences are related and that we have the correct tree. The internal nodes represent unknown ancestor sequences.

Definition 1.1. *Given is a set of sequences* $S = \{s_1, .., s_n\}$ *with* $s_i \in \Sigma^*$ *where* Σ *is a finite alphabet. A* Multiple Sequence Alignment (MSA) *consists of a set of sequences* $\mathrm{A} = \langle a_1, a_2, .., a_n \rangle$ *with* $a_i \in {\Sigma'}^*$ *where* $\Sigma' = \Sigma \cup \{``\_"\} \notin \Sigma$. $\forall a_i \in \mathrm{A} : |a_i| = k$. *The sequence obtained from* $a_i \in \mathrm{A}$ *by removing all "_" gap characters is equal to* s_i.

Definition 1.2. *The character "_" or any contiguous sequence of "_" within an aligned sequence* $a_i \in \mathrm{A}$ *is called a* gap. *A gap corresponds to an insertion or deletion event* (indel).

Definition 1.3. *The* tree $T(S) = (V, E, S)$ *is a binary, leaf-labeled tree with leafset* $S = \{s_1, .., s_n\}$.

Definition 1.4. *A* tree scoring function *is a function* $F : T \to \mathbb{R}$.

In our context a *tree* $T(S)$ associated with a set of sequences $S = \{s_1, .., s_n\}$ is the tree that corresponds to the evolutionary history of the sequences of S. The internal nodes V represent (usually unknown) ancestor sequences.

Constructing trees, MSAs, and ancestral sequences, one encounters three sorts of problems. First, all are dependent on a model for evolutionary processes. At the level of the protein (which is our exclusive concern here), modeling the evolutionary history of a family must begin with assumptions about the frequencies of amino acid mutations, insertions, and deletions. These frequencies are now available from empirical studies of protein sequences (Gonnet *et al.*, 1992). In this work, a simple model, derived originally from work of Dayhoff *et al.* (1978) and subsequently amplified (Brenner *et al.*, 1993) is used.

Second, constructing models for evolutionary histories encounters problems of mathematical complexity. Most versions of the MSA problem (Carillo and Lipman, 1988; Gupta *et al.*, 1995, 1996; Kececioglu, 1993; Wang and Gusfield, 1996; Roui and Kececioglu, 1998; Jiang *et al.*, 1996; Sankoff and Cedergren, 1983) and the exact construction of evolutionary trees (Sankoff, 1975; Dress and Steel, 1993; Estabrook *et al.*, 1975, 1976; Felsenstein, 1973, 1981; Thorne *et al.*, 1993) is NP-complete (Foulds and Graham, 1982; Jiang and Wang, 1994), and becomes computationally expensive for many real protein families encountered in a contemporary database. This requires that virtually all MSAs and evolutionary trees that will be used in the post-genomic era will be constructed using approximate heuristics.

Definition 1.5. *An* MSA scoring function *is a function* $F : \mathrm{A} \to \mathbb{R}$.

Definition 1.6. *Let* $\mathcal{A}$ *be the set of all possible MSAs that can be generated for a given set of sequences* $S = \{s_1, s_2, .., s_n\}$. *The* optimal MSA $\bar{\mathrm{A}} \in \mathcal{A}$ *is an MSA such that w.l.o.g.* $F(\bar{\mathrm{A}}) = \max_{\mathrm{A} \in \mathcal{A}} F(\mathrm{A})$. *In some scoring functions the minimum is the optimal.*

Problem 1.1 (MSA problem). Given is a set of sequences $S = \{s_1, .., s_n\}$. Find the *optimal MSA* A for S.

The use of heuristics in constructing MSAs creates a third problem, one centering on evaluation. Before using an MSA or tree built by a heuristic, one would like to know approximately how closely the heuristic has approximated an optimum MSA or tree. Even today, it is common for biochemists to evaluate by eye (and adjust by hand) the output of MSA tools. This is a clearly inadequate approach for any systematic

```
A: RPCVCP___VLRQAAQ__QVLQRQIIQGPQQLRRLF_AA
B: RPCACP___VLRQVVQ__QALQRQIIQGPQQLRRLF_AA
C: KPCLCPKQAAVKQAAH__QQLYQGQLQGPKQVRRAFRLL
D: KPCVCPRQLVLRQAAHLAQQLYQGQ____RQVRRAF_VA
E: KPCVCPRQLVLRQAAH__QQLYQGQ____RQVRRLF_AA
```

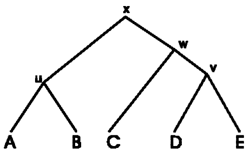


FIG. 1. The sequences $\{A, B, C, D, E\}$ of the MSA on the left are related by the evolutionary tree on the right.

reconstruction of natural history using genomic sequence data. A formal method for judging the quality of an MSA is needed.

Accordingly, a variety of groups have proposed or used scoring functions (Sankoff and Cedergren, 1983; Altschul, 1989; Thompson *et al.*, 1994; Higgins and Sharp, 1989; Carillo and Lipman, 1988; Gupta *et al.*, 1996) that assess the quality of an MSA. In this paper, we are interested in MSAs when no tree is available. In this case, the most commonly used function follows a simple approach that examines every pair of proteins in the family, generates a score for each pairwise alignment using a Dayhoff matrix (Dayhoff *et al.*, 1978), and creates a score for the MSA by summing each of the scores of the pairwise alignments. We shall call these "sum of pairs" (SP) methods.

1.1. Scoring pairwise sequence alignments

To determine if two sequences $s_1, s_2 \in \Sigma$ are related and have a common ancestor, the sequences are usually aligned, and the problem is to find the alignment that maximizes the probability that the two sequences are related.

To actually calculate these probabilities, one applies a Markovian model for sequence evolution (Krogh *et al.*, 1994; Baldi *et al.*, 1994). This begins with an alignment of the two sequences, e.g., as follows.

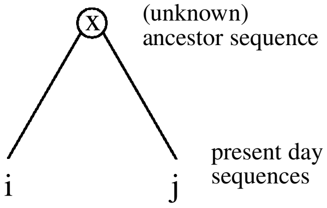
```
VNRLQQNIVSL____________EVDHKVANYKPQVEPFGHGPIFMATALV PGLYLLPL
VNRLQQSIVSLRDAFNDGTKLLEELDHRVLNYKPQANPFGNGPIFMVTAIV PGLHLLPI
```

The gaps arise from *insertions* (or their counterpart *deletions*) during divergent evolution. The alignment is normally done by a dynamic programming (DP) algorithm using Dayhoff matrices (Gotoh, 1982; Smith and Waterman, 1981; Needleman and Wunsch, 1970; Altschul and Erickson, 1986), which finds the alignment that maximizes the probability that the two sequences evolved from an ancestral sequence as opposed to being random sequences. An affine gap cost is used according to the formula $a + l \cdot b$, where a is a fixed gap cost, l is the length of the gap and b is the incremental cost (Altschul and Erickson, 1986; Brenner *et al.*, 1993). More precisely, we are comparing two possibilities:

a) that the two sequences arose independently of each other (implying that the alignment is entirely arbitrary, with the alignment of amino acid i in one protein to amino acid j in the other occurring no more frequently than expected by chance, which is equal to the product of the individual frequencies with which amino acids i and j occur in the database)

$$Pr\{i \text{ and } j \text{ are independent}\} = f_i f_j, \tag{1}$$

b) that the two sequences have evolved from some common ancestral sequence after t units of evolution where t is measured in PAM units (Gonnet, 1994b).



$$Pr\{i \text{ and } j \text{ descended from some } x\} = \sum_x f_x Pr\{x \rightarrow i\} Pr\{x \rightarrow j\} \tag{2}$$

Definition 1.7. *A* 1-PAM *unit is the amount of evolution which will change, on average, 1% of the amino acids. In mathematical terms, this is expressed as a matrix M such that*

$$\sum_{i=1}^{20} f_i(1 - M_{ii}) = 0.01$$

where f_i is the frequency of the i^{th} amino acid.

Definition 1.8. *The score of an* optimal pairwise alignment $OPA(s_1, s_2)$ *of two sequences* s_1, s_2 *is the score of an alignment with the* maximum score *where a probabilistic scoring method (Dayhoff* et al.*, 1978; Gonnet* et al.*, 1992) is used. We refer to a pairwise alignment of two sequences* s_1, s_2 *with* $\langle s_1, s_2\rangle$.

$$D_{ij} = 10\log_{10}\left(\frac{Pr\{i \text{ and } j \text{ descended from some } x\}}{Pr\{i \text{ and } j \text{ are independent}\}}\right) \tag{3}$$

The entries of the Dayhoff matrix are the logarithm of the quotient of these two probabilities. Note that scores represent the probabilities that the two sequences have a common ancestor. The larger the score is, the more likely it is that the two sequences are homologous and therefore have a common ancestor.

1.2. Sum-of-pairs measure

Definition 1.9. *The score of the induced MSA-derived pairwise alignment* MPA($\mathrm{a_i}$, $\mathrm{a_j}$) *of two sequences* $a_i, a_j \in \mathrm{A}$ *is the score of the alignment of the two strings* a_i *and* a_j*. Opposing gaps are removed.*

To calculate the score with the SP measure (Carillo and Lipman, 1988), all $\binom{n}{2}$ scores of the pairwise alignments within the MSA (see Definition 1.9) are added up. SP methods are obviously deficient from an evolutionary perspective (Altschul and Lipman, 1989). Consider a tree (Figure 2) constructed for a family containing five proteins. The score of a pairwise alignment $\langle A, B\rangle$ evaluates the probability of evolutionary events on edges (u, A) and (u, B) of the tree; that is, the edges that represent the evolutionary distance between sequence A and sequence B. Likewise, the score of a pairwise alignment $\langle C, D\rangle$ evaluates the likelihood of evolutionary events on edges (C, w), (w, v) and (v, D) of the tree.

By adding to the evolutionary tree "ticks" that are drawn each time an edge is evaluated when calculating the SP score (Figure 2), it is readily seen that, with the SP method, different edges of the evolutionary tree of the protein family are counted a different numbers of times. In the example tree on the left side that corresponds to the MSA of Figure 1, edges (r, u), (r, w) and (w, v) are each counted six times by the SP method, while edges (u, A), (u, B), (v, D), (v, E), and (w, C) are each counted four times. The numbers on the edges are the numbers of "ticks." It gets worse as the tree grows (see tree on the right). There is no theoretical justification to suggest that some evolutionary events are more important than other ones.

Thus, SP methods are intrinsically problematic from an evolutionary perspective for scoring MSAs. This was the motivation to develop a scoring method that evaluates each edge equally. In addition, we wanted a scoring function that does not depend on the actual tree structure.

We report here a "circular sum" (or CS) method for evaluating the quality of an MSA. The method uses a solution to the Traveling Salesman Problem, which identifies a circular tour through an evolutionary tree connecting the sequences in a protein family. The algorithm gives an upper bound, the best score that can possibly be achieved by any MSA for a given set of protein sequences. Both the bound and the circular tour can be derived without explicit knowledge of the correct evolutionary tree; thus, the method can be applied without need to address the mathematical issues involved in tree construction. Last, it gives us an absolute score of MSAs, which is important in designing and verifying MSA heuristics.

Both the tree construction problem and the Traveling Salesman Problem are NP complete. But the Traveling Salesman Problem (see next section) has been studied very extensively (Johnson, 1987, 1990), and optimal solutions can be calculated within a few hours for up to 1000 cities and in a few seconds for up to 100 cities, whereas the construction of evolutionary trees is still a big problem. There are heuristics for large scale problems that calculate near optimal solutions that are within 1% to 2% of the optimum (Padberg and Rinaldi, 1991; Groetschel and Holland, 1991).

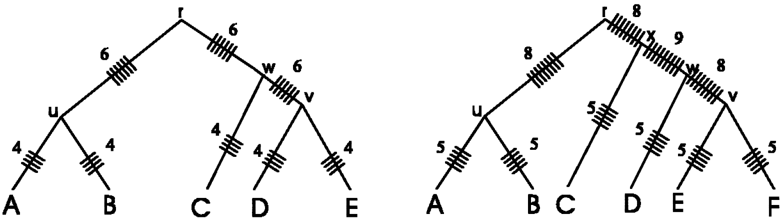


FIG. 2. Traversal of a tree using the SP measure. Some edges are traversed more often than others. The numbers on the edges represent the number of "ticks."

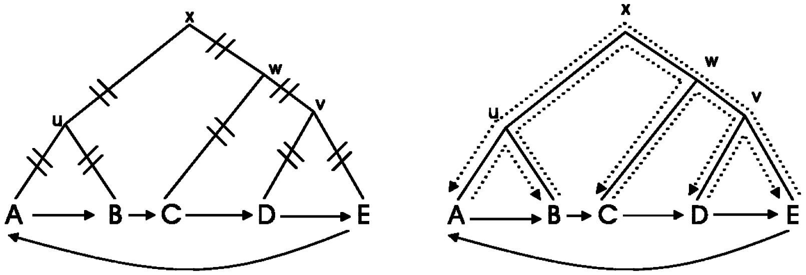


FIG. 3. Traversal of a tree in circular order.

2. METHODS

The task of generating an evolutionary unbiased score for an MSA can be rephrased as a simple problem: How can a tree be traversed, going from leaf to leaf, such that all edges (each representing its own episode of evolutionary divergence) are counted the same number of times? Assume that a tree contains 5 leaves, labeled from A to E.

2.1. Circular tours

Definition 2.1. *A* Circular order *$C(T)$ of the set of sequences $S = \{s_1, .., s_n\}$ is any tour through a tree $T(S)$ where each edge is traversed exactly twice and each leaf is visited once.*

The problem can be solved by walking through the tree in a circular order, that is, from leaf A to B, from B to C, from C to D, from D to E and then back from E to leaf A (Figure 3); all edges are counted exactly twice, independent of the tree structure.

Lemma 2.1 (Shortest Tour). *The circular tour is the shortest possible tour through a tree that visits each leaf once (see Figure 3). It traverses all edges exactly twice and thus weights all edges of the evolutionary tree equally.*

Proof 2.1. Starting with 2 leaves, the proof is obvious: there is only one tour and all edges are counted exactly twice (see Figure 4). ■

Definition 2.2. *A* subtree *$T_u(V', E')$ of $T(V, E)$ is a tree T_u with $V' \subset V$, $E' \subset E$, where u is the root of T_u and $u \in V'$, and all the directed paths from u to the leaves in T are also present in T_u.*

For a tree with n leaves do the following: choose any edge x in the tree and label the four subtrees (see Definition 2.2) attached to x as A, B, C and D (see Figure 5). We now look at all possible circular tours for the tree, starting with subtree A and ignoring the tours within the subtrees themselves. There are four circular tours ($[A, B, C, D]$, $[A, B, D, C]$ and the reverse tour of both). All circular tours traverse edge x exactly twice.

Since the division into subtrees can be done anywhere in the tree, all edges are counted twice. There is no shorter tour, because we have to come back to the first subtree A.

Lemma 2.2 (Noncircular Tours). *Any other noncircular tour traverses at least one edge more than twice.*

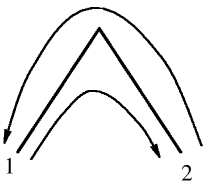


FIG. 4. Circular tour for 2 leaves.

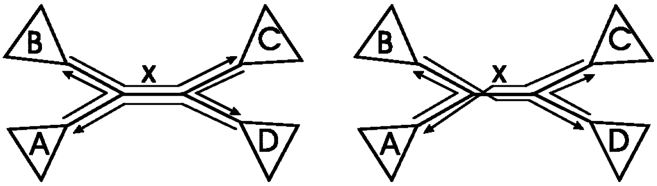


FIG. 5. Circular tours with respect to edge x.

Proof 2.2. Choose any edge x in the tree, and label the four subtrees attached to x with A, B, C and D (see Figure 6). We now look at all possible noncircular tours for the tree, starting with subtree A, and ignoring the tours within the subtrees themselves. There are two noncircular tours ($[A, D, B, C]$ and $[A, C, B, D]$). In this case, edge x in the middle is traversed four times in both cases. Since this can be done for any edge x in the tree, the conclusion is that any noncircular tour traverses at least one edge more than twice. ■

Definition 2.3. *Given is a tree T. An* isomorphic *tree T' of T is a tree with the same tree topology as T, but it may have a different graphical representation, where one or many subtrees may be rotated.*

Trees can be drawn in many different ways, as each subtree can be rotated. If we count how many different orderings of the leaves there are such that we still end up with the same tree topology, then if we always start with the same leaf (if we ignore the breaking of the circularity), there are $2^{(n-2)}$ such possible ways to draw a tree.

Lemma 2.3 (Isomorphic Trees). *A circular tour $C(T)$ of a tree T is also a circular tour for all isomorphic trees of T.*

Proof 2.3. Take the same tree as in Figure 5, but this time rotate the subtrees A, B (swap labels A and B). When the tree is traversed again in the order $[A, B, C, D]$ and back to A, the middle edge is again only traversed twice, the same accounts for the other circular orders. Again, since the division into subtrees can be done anywhere in the tree, all edges are counted twice, therefore this is true for all isomorphic trees. ■

An immediate conclusion from this is that any tree has $2^{(n-2)}$ different circular tours, ignoring the circularity (we always start with the same leaf).

Definition 2.4. *A tour C_i is* shorter *than a tour C_j ($C_i < C_j$) if the sum of edge lengths that are traversed by C_i is smaller than the sum of edge lengths that are traversed by C_j.*

Definition 2.5. *Let L be a set of n leaves of a tree T. The* distance *between two leaves $x, y \in L$ is δ_{xy}. It is the unique path length (the sum of all edges in PAM distances) from leaf x to leaf y. We assume that distances are symmetric, hence $\delta_{xy} = \delta_{yx}$.*

Definition 2.6. *Given is a circular tour C of length n of a tree T. The* path length *$P(T)$ of a tree T is then $P(T) = \frac{1}{2}\sum_{i=1}^{n} PAM(s_{C_i}, s_{C_{i+1}})$, where $C_{n+1} = C_1$.*

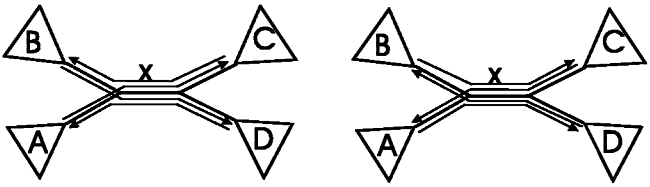


FIG. 6. Noncircular tours with respect to edge x.

Example 2.1. The order $\langle A, B, C, D, E\rangle$ is a circular order in Figure 7, but not $\langle A, C, B, D, E\rangle$. In the second example, edges $\langle x, u\rangle$ and $\langle x, w\rangle$ are counted four times, while all other edges are counted only twice.

2.2. Traveling salesman problem application

The PAM distances derived from pairwise alignments are now the key to identifying a circular tour. For a set of protein sequences, it is computationally simple to obtain a set of $\binom{n}{2}$ pairwise PAM distances by aligning each sequence with every other sequence using a dynamic programming algorithm to obtain the Optimal Pairwise Alignment. Our goal is to be able to find a circular tour without the need of constructing an evolutionary tree.

It is easy to find a circular order of a given tree T. But in reality we do not have a tree. The only information available to us is just the sequences and the PAM distances. The question then is: how can we find a circular order of the tree?

To answer this question, we need to think of the following:

- We are only interested in the tree that corresponds to the measured distances of the pairwise alignments.
- This tree has circular orders.
- A circular order is the shortest tour through a tree.
- A shortest tour can be calculated using a TSP algorithm.

Problem 2.1 (Sequence TSP problem). Given is a set of sequences $S = \{s_1, .., s_n\}$ and the corresponding $\binom{n}{2}$ *PAM* distances of the optimal pairwise alignments. The problem is to find the shortest tour where each sequence is visited once.

Definition 2.7. *The* TSP order $CS(S)$ *of a set of sequences* $S = \{s_1, .., s_n\}$ *is the order of the sequences that is derived from the optimal solution of a TSP.*

Hence, to find such a circular tour, we need to find the shortest tour from leaf to leaf of the given set of sequences S. To solve this problem we reduce it to the symmetric Traveling Salesman Problem (TSP): given is a matrix M that contains the $\binom{n}{2}$ distances of n cities (Johnson, 1987, 1990). The problem is to find a tour (each city is visited once), where the overall path must be minimal. We use a modified version of the problem: in our case, the cities correspond to the sequences and the distances are the PAM distances of the pairwise alignments. The TSP problem is known to be NP complete (Papadimitriou, 1977), but it is very well studied and optimal solutions can be calculated within a few hours for up to 1000 cities and in a few seconds for up to 100 cities. There are heuristics for large scale problems that calculate near optimal solutions that are within 1% to 2% of the optimum (Padberg and Rinaldi, 1991; Groetschel and Holland, 1991). For real applications we seldom have more than 100 sequences to compare simultaneously, and the calculation of the optimal TSP solution usually takes only a small fraction of the time it takes to compute all pairwise alignments to derive the PAM distances. With $P(S)$ we will denote the path length of a tree that corresponds to the sequences S.

Definition 2.8. *Let C be the output permutation of a TSP algorithm for a set of sequences* $S = \{s_1, s_2, .., s_n\}$*, then the* path length $P(S)$ *for* S *is:* $P(S) = \frac{1}{2}\sum_{i=1}^{n} PAM(s_{C_i}, s_{C_{i+1}})$*, where* $C_{n+1} = C_1$.

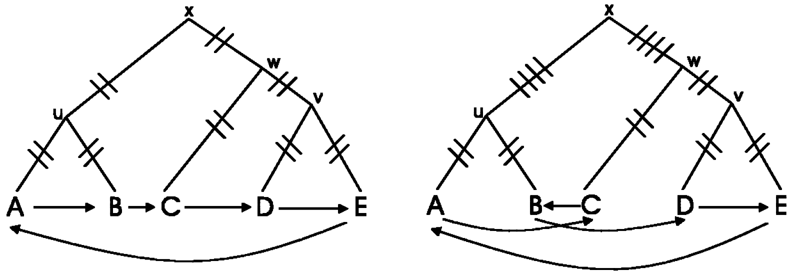


FIG. 7. Traversal of a phylogenetic tree in a circular $\langle A, B, C, D, E\rangle$ and noncircular order $\langle A, C, B, D, E\rangle$.

The path length $P(T)$ for a given tree T is defined in a similar way, except that the circular order does not need to be determined but can be derived directly from the evolutionary tree.

2.3. Scoring of MSAs without evolutionary trees

The CS algorithm provides us with a circular tour of the unknown evolutionary tree. This tour can now be used to evaluate a specific MSA or to compare different algorithms for constructing MSAs.

For pairwise alignments, usually the score of the optimal pairwise alignment (see Definition 1.8) is used.[1] For scoring an MSA though, we will obviously not use OPA scores, but scores *derived* from the alignments within the MSA itself (OPA scores are used to determine the upper bound, see below). The pairwise alignments within the MSA are scored using a Dayhoff matrix without dynamic programming, as the sequences in an MSA are already aligned. We will call these scores MPA scores, a shortcut for scores of MSA-derived pairwise alignments (see Definition 1.9).

Definition 2.9. *We can now extend the definition of the function $s(x, y)$, which scores two symbols $x, y \in \Sigma'$:*

$$s(x,y) = \begin{cases} D_{xy},\ \textit{if } x \neq \text{“\_”} \textit{ and } y \neq \text{“\_”} \\ 0 \textit{ if } x = \text{“\_”} \textit{ and } y = \text{“\_”} \\ \textit{otherwise an affine gap cost that depends on the gap length}^2 \end{cases}$$

(Altschul and Erickson, 1986; Brenner et al.*, 1993). D_{xy} is an entry in a Dayhoff matrix (Schwarz and Dayoff, 1979; Gonnet* et al.*, 1992).*

The function $s(x, y)$ scores two symbols x, y that are either amino acids or gap characters. If there is a deletion in both sequences, there is no penalty (score is zero) because that deletion happened in some ancestor, and its penalty has been counted already .

The MPA scores are not larger than the scores obtained from the optimal pairwise alignments (OPA). The CS score of an MSA is the sum of MPA scores in the circular order C divided by two.

Definition 2.10. *The score $CS(\mathrm{A})$ of an MSA* A *is defined as: $CS(\mathrm{A}) = \frac{1}{2}\sum_{i=1}^{n} MPA(a_{C_i}, a_{C_{i+1}})$ where $C_{n+1} = C_1$ and where C is a circular order.*

Definition 2.11. *The* upper bound *$CS_{max}(S)$ for a set of sequences S is defined as: $CS_{max}(S) = \frac{1}{2}\sum_{i=1}^{n} OPA(a_{C_i}, a_{C_{i+1}})$ where $C_{n+1} = C_1$ and where C is a circular order.*

Definition 2.12. *The* optimal score *$CS_{opt}(S)$ for a set of sequences S is defined as follows. Let $\mathcal{A}$ be the set of all possible MSAs that can be generated for a given set of sequences $S = \{s_1, s_2, .., s_n\}$: $CS_{opt}(S) = \max_{\mathrm{A}\in\mathcal{A}} CS(\mathrm{A})$.*

Lemma 2.4 (Upper Bound). *Let $\mathcal{A}$ be the set of all possible MSAs that can be generated for a given set of sequences $S = \{s_1, s_2, .., s_n\}$. The maximal score $CS_{max}(S)$ serves as an upper bound for the score of any MSA* A *that can be built from S: $CS_{max}(S) \geq \max_{\mathrm{A}\in\mathcal{A}} CS(\mathrm{A})$.*

Proof 2.4. Let us build two matrices A and B, one matrix A with the $\binom{n}{2}$ *OPA* scores of the sequences S and a matrix B with the $\binom{n}{2}$ *MPA* scores. Since *OPA* scores are never smaller than *MPA* scores, each entry in matrix A is as least as large as the corresponding entry in matrix B. Hence whatever tour C we choose, the sum of the scores of matrix A will always be at least as large as the sum of the scores in matrix B. ■

[1] A larger score corresponds to a smaller PAM distance, as we will show in Appendix A.

[2] The gaps are scored according to the formula $a + l \cdot b$, where a is a fixed gap cost that depends on the PAM distance, l is the length of the gap and b is the incremental cost which also depends on the PAM distance (Benner *et al.*, 1993).

```
A: RPCVCP___VLRQAAQ__QVLQRQIIQGPQQLRRLF_AA
B: RPCACP___VLRQVVQ__QALQRQIIQGPQQLRRLF_AA
C: KPCLCPKQAAVKQAAH__QQLYQGQLQGPKQVRRAFRLL
D: KPCVCPRQLVLRQAAHLAQQLYQGQ____RQVRRAF_VA
E: KPCVCPRQLVLRQAAH__QQLYQGQ____RQVRRLF_AA
```

	A	B	C	D	E
A	0	336	27	44	110
B	336	0	79	56	99
C	27	79	0	171	176
D	44	56	171	0	327
E	110	99	176	327	0

FIG. 8. An MSA and a table containing the pairwise scores.

Whenever we score an MSA, regardless of the algorithm used to generate it, the score of this alignment must be less than or equal to the upper bound $CS_{max}(S)$. Hence the upper bound can be used to evaluate a given MSA. The closer the score of a calculated MSA is to the upper bound, the better the MSA is.

2.4. *Optimal Score and Upper Bound*

The optimal score of an MSA, which is the MSA with the maximum *possible* score, and the upper bound are not necessarily the same. There are two cases:

- $CS_{opt}(S) = CS_{max}(S)$. The upper bound is equal to the optimal score. Whenever the MPA scores of the optimal MSA and the OPA scores are equal, then the upper bound is equal to the optimal score. For instance, if an MSA does not have any gaps, then obviously the optimal score is equal to the upper bound, as the MPA and OPA scores are equal. See also Example 2.2.
- $CS_{opt}(S) < CS_{max}(S)$. The upper bound is larger than the optimal score. This is the case when the MPA scores of the optimal MSA are smaller than the OPA scores.

In general, whenever $CS(\mathrm{A}) = CS_{max}$, when the score of the MSA is equal to the upper bound, we know that the MSA is optimal.

Example 2.2. The table in Figure 8 shows the pairwise MPA scores of the sequences from the MSA on the left. It is easy to verify that the tour that yields the highest score is (A, B, C, D, E, A) and that it is a circular tour in the corresponding tree. The score of the MSA is the sum of pairwise alignments in circular order divided by two, so $F(\mathrm{A}) = (336 + 79 + 171 + 327 + 110)/2 = 512$. Here the MSA has the optimal score, $CS(\mathrm{A}) = CS_{max} = CS_{opt}$.

We have a way to determine a circular order and an upper bound on the score of an MSA for a set of sequences. This score has been obtained without requiring the calculation of an evolutionary tree. Only the sequences at the leaves and their PAM distances/scores with each other are needed.

3. SIMULATION OF EVOLUTION

To illustrate how the scoring function can be used, a variety of tools for generating MSAs were challenged with a set of protein families simulated following a Markovian model of evolution, and the outputs of each were evaluated using the CS measure. This provides, of course, only an approximate assessment of the MSA tools themselves. A better assessment must come with actual experimental sequence data.

Random trees with a given structure and edge lengths and a random sequence at the root were generated. From this, sequence mutations, insertions and deletions of different sizes were introduced according to the length of the edges of the tree. At each internal node, a new sequence was thus generated. At the end of the simulation, only the sequences at the leaves are retained. Since both the places of insertions and deletions, as well as the real tree are known, the correct MSA is known as well.

The retained sequences at the leaves can be given to different algorithms: MSA (Gupta *et al.*, 1996; Lipman *et al.*, 1989), MAP (Huang, 1994), ClustalW (Higgins and Sharp, 1989; Thompson *et al.*, 1994) and the Probabilistic model (PAS) (Gonnet and Brenner, 1996; Gonnet, 1994a). Also, the score of the calculated MSAs can be compared to the score of the "real" (generated) MSA using the CS measure.

The results for 3 combinations of trees and sizes are shown. These are representative of all other results (Figures 10–12). The circular order was always derived using a TSP algorithm, not with the generated tree. But since we have the correct tree, it was easy to verify that the TSP order is in fact a circular order (which was always the case).

3.1. Example

From a random evolutionary tree with 8 leaves, a random MSA was produced. That is, the original sequence was mutated and in this case there were two indel events.

The score of the alignment is the MPA score introduced in Definition 2.10. It is the score derived from the pairwise alignments *within* the MSA, in TSP order. In this example the order is 1, 2, 6, 5, 7, 8, 3, 4. You can easily verify that this tour is a circular tour of the tree in Figure 9. The numbers on the edges of the tree are PAM distances.

The maximum possible score, the $CS_{max}(S)$ score (see Definition 2.8), is slightly better than the score derived from the alignment (using MPA scores). This means that the real alignment is not optimal, and the gaps could probably be shifted a bit to either side to increase the score. The indel events are one deletion event that happened in the ancestor of sequence 3 and 4 and one insertion event that happened in the ancestor of sequences 1 and 2.

Note that when the gaps are scored, each indel event is scored only once. When sequence 8 is aligned against sequence 3, the gap is scored. But when the alignment of sequences 3 and 4 is scored in the MSA, the gap scores 0, because it has already been accounted for, and you could actually simply remove the gap in the MPA (see Definition 1.9), since they have exactly the same length. Another observation is that the gaps form "blocks," that is, the gaps that belong to the same evolutionary event are not interrupted with sequences without gaps.

The sequences at the leaves are then fed to different algorithms. None of the algorithms knows the correct tree or the correct MSA.

```
Generated random MSA:
--------------------------------------------
Score of the alignment (MPA): 1311.907
Maximum possible score (OPA): 1328.206

1       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGA KLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGA KLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGSVYLR___HVEIDLRTQVGSSPYAATVIKNVIKNT RPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGSVILK___HLEIDFQSSVGSNPRAATYVKNVIKGR KPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGSVLLR___TLELDFQGQVGTPPRAAVYVKNVTKGA KPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGSVVLR___HFELDFEGQAETNPRSSVYVKNVSQGV EPIHLTEWQEFNYGNVSCKI
3       LDVLDVTT___________IIIQ___TFRIDLQEQLGSNPASATYVKNILTGA KLLHLSEGEEYTMGHAVLIM
4       LDVLDVVT___________IIVV___TFRIDLQEQVGENPASASYVQDILTGA KLLHLSDGKEYTMGHVVAII
```

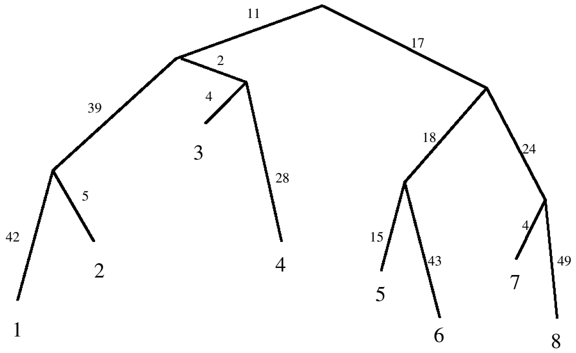


FIG. 9. Tree associated with example MSA.

```
Sequences:
----------
1 LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNE LTGAKLLHISDGQDTTIGQTVAIT
2 LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNK LTGAKLLHLSDGQESAMGQVVAIT
3 LDVLDVTTIIIQTFRIDLQEQLGSNPASATYVKNILTGAKLLHLSEGEE YTMGHAVLIM
4 LDVLDVVTIIVVTFRIDLQEQVGENPASASYVQDILTGAKLLHLSDGKE YTMGHVVAII
5 LHVIDERKRLPAARFNKGSVILKHLEIDFQSSVGSNPRAATYVKNVIKG RKPLKLCDGQEISLGLIVCIW
6 LHVIDERKRLEAARFNKGSVYLRHVEIDLRTQVGSSPYAATVIKNVIKN TRPLKLCMGQELSLGMIVMLF
7 LAVIEVRRGQVALEFNKGSVLLRTLELDFQGQVGTPPRAAVYVKNVTKG AKPLHLVEGQEFNLGYVTCII
8 VHVADVTRGLTRLEFDKGSVVLRHFELDFEGQAETNPRSSVYVKNVSQG VEPIHLTEWQEFNYGNVSCKI

TSP ordering of sequences:
--------------------------
[1, 2, 6, 5, 7, 8, 3, 4, 1]
```

First, the sequences were fed to the MSA algorithm, which produces the following output:

```
MSA:
---
Score of the alignment (MPA): 1305.050
Maximum possible score (OPA): 1328.206

1       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGA KLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGA KLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGSVYLR___HVEIDLRTQVGSSPYAATVIKNVIKNT RPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGSVILK___HLEIDFQSSVGSNPRAATYVKNVIKGR KPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGSVLLR___TLELDFQGQVGTPPRAAVYVKNVTKGA KPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGSVVLR___HFELDFEGQAETNPRSSVYVKNVSQGV EPIHLTEWQEFNYGNVSCKI
3       LDVLDV___________TTIIIQ___TFRIDLQEQLGSNPASATYVKNILTGA KLLHLSEGEEYTMGHAVLIM
4       LDVLDV___________VTIIVV___TFRIDLQEQVGENPASASYVQDILTGA KLLHLSDGKEYTMGHVVAII
```

The alignment looks very similar to the constructed MSA. The only difference is that the gaps appear in different places, and the score is slightly lower than the score of the constructed MSA. In the simulation, the difference of the upper bound and the CS score was noted (see tables). In this case, the difference would be 23.156.

The next algorithm is the probabilistic model. In this case the algorithm merges the two gaps:

```
Probabilistic model:
-------------------
Score of the alignment (MPA): 1305.080
Maximum possible score (OPA): 1328.206

1       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGA KLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGA KLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGS___VYLRHVEIDLRTQVGSSPYAATVIKNVIKNT RPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGS___VILKHLEIDFQSSVGSNPRAATYVKNVIKGR KPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGS___VLLRTLELDFQGQVGTPPRAAVYVKNVTKGA KPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGS___VVLRHFELDFEGQAETNPRSSVYVKNVSQGV EPIHLTEWQEFNYGNVSCKI
3       LDVLDVTT______________IIIQTFRIDLQEQLGSNPASATYVKNILTGA KLLHLSEGEEYTMGHAVLIM
4       LDVLDVVT______________IIVVTFRIDLQEQVGENPASASYVQDILTGA KLLHLSDGKEYTMGHVVAII
```

The last algorithm tested in this example is ClustalW. It looks like the algorithm also found two indel events, but when you look closer you can see that the gaps are shifted against each other. This means that the second block of gaps are two indel events and not just one, which is the reason for the lower score.

```
ClustalW:
---------
Score of the alignment (MPA): 1291.417
Maximum possible score (OPA): 1328.206

1       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGA KLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGA KLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGSVYLR___HVEIDLRTQVGSSPYAATVIKNVIKNT RPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGSVILK___HLEIDFQSSVGSNPRAATYVKNVIKGR KPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGSVLLR___TLELDFQGQVGTPPRAAVYVKNVTKGA KPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGSVVLR___HFELDFEGQAETNPRSSVYVKNVSQGV EPIHLTEWQEFNYGNVSCKI
3       LDVLDVTT___________III___QTFRIDLQEQLGSNPASATYVKNILTGA KLLHLSEGEEYTMGHAVLIM
4       LDVLDVVT___________IIV___VTFRIDLQEQVGENPASASYVQDILTGA KLLHLSDGKEYTMGHVVAII
```

Following is the result of a large simulation. For each tree type, hundreds of alignments were produced. This is the reason why the scores have variances, because the upper bound is not the same for each set of sequences, but lies in the same order, because similar PAM distances were used and the same

Method	*CS score*	*Upper bound − CS score*
Upper bound:	22451 ± 516	0
MSA:	21767 ± 513	684 ± 76
PAS	21414 ± 481	740 ± 74
Real:	21367 ± 439	781 ± 74
ClustalW:	21330 ± 473	822 ± 76
MAP:	18633 ± 608	3521 ± 583
Dummy:	5948 ± 911	16200 ± 826

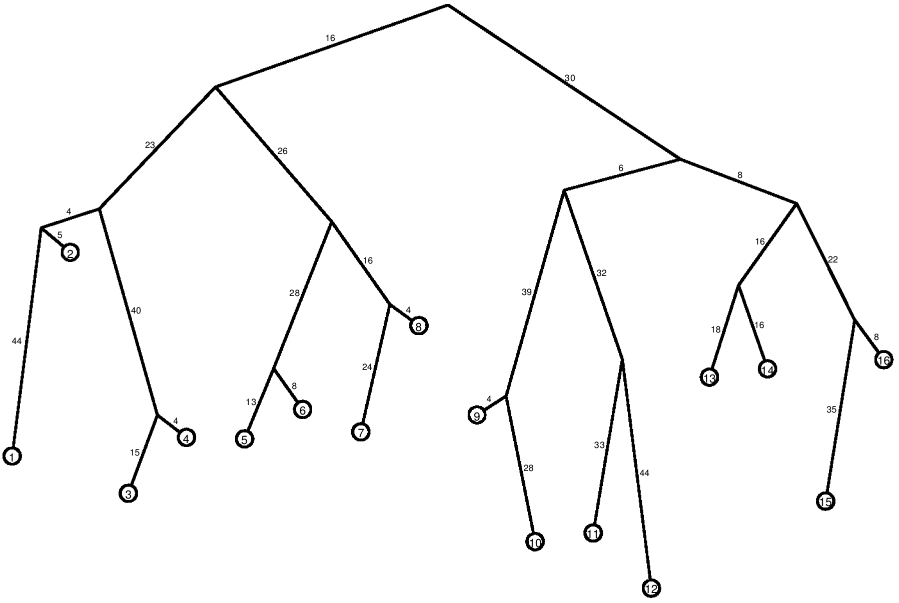


FIG. 10. Comparison of different MSA methods: the CS score (second column) is calculated using a TSP ordering. The upper bound is the CS score based on the optimal pairwise alignment.

underlying tree structure. Higher scores or smaller differences mean better alignments. The rows are sorted into ascending order.

Figure 10 shows the result for balanced binary trees with 16 leaves. The length of the sequences is 300 amino acids and the average edge distance is 30 PAM (so the maximum PAM distance between two sequences is about 240 PAM).

For this tree, MSA scored the best followed by PAS, ClustalW and MAP. The alignments of MSA and PAS are slightly better than the "real" alignment. The reason for this is that the simulated MSAs are not necessarily optimal. As a comparison, the score of a bad alignment (all the sequences aligned without deletions) was calculated in the last row (Dummy). The tree below is an example of a generated phylogenetic tree.

The next trees (Figure 11) are unbalanced with 30 sequences of length 300 and the average edge distance is 30 PAM (so the maximum PAM distance is about 300). Only PAS, MAP and ClustalW were able to compute the alignments (in a reasonable time). In this case, PAS did slightly better than ClustalW.

The trees of the last table shown here (Figure 12) had 50 sequences with length 300 and an average edge length of 15 PAM. In the simulation, the PAS method was the best, followed by ClustalW. No other methods were able to calculate an MSA in a reasonable time.

These experiments show how the CS measure can be used. It also shows that the upper bound $CS_{max}(S)$ is very close to the actual score CS(A) of the simulated MSA. Obviously the ultimate evaluation of tools must be done with real rather than simulated data.

Method	*CS score*	*Upper bound − CS score*
Upper bound:	40282 ± 581	0
Real:	38796 ± 573	1486 ± 138
PAS	38638 ± 594	1643 ± 151
ClustalW:	38465 ± 597	1818 ± 153
MAP	21238 ± 1565	18957 ± 1507
Dummy:	10609 ± 1321	28641 ± 1308

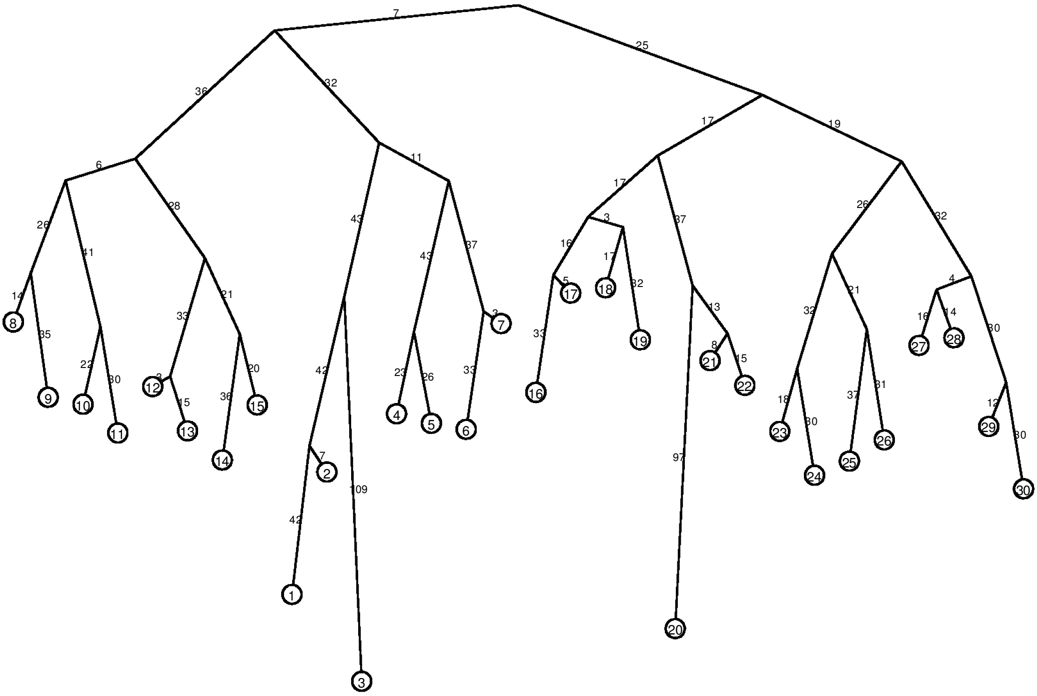


FIG. 11. Comparison of different MSA methods: the CS score (second column) is calculated using a TSP ordering. The upper bound is the CS score based on the optimal pairwise alignment.

Method	*CS score*	*Upper bound − CS score*
Upper bound:	91232 ± 924	0
Real:	88960 ± 913	2272 ± 167
PAS:	88494 ± 941	2704 ± 175
ClustalW:	87313 ± 1654	3884 ± 1363
Dummy:	34062 ± 4558	57456 ± 3954

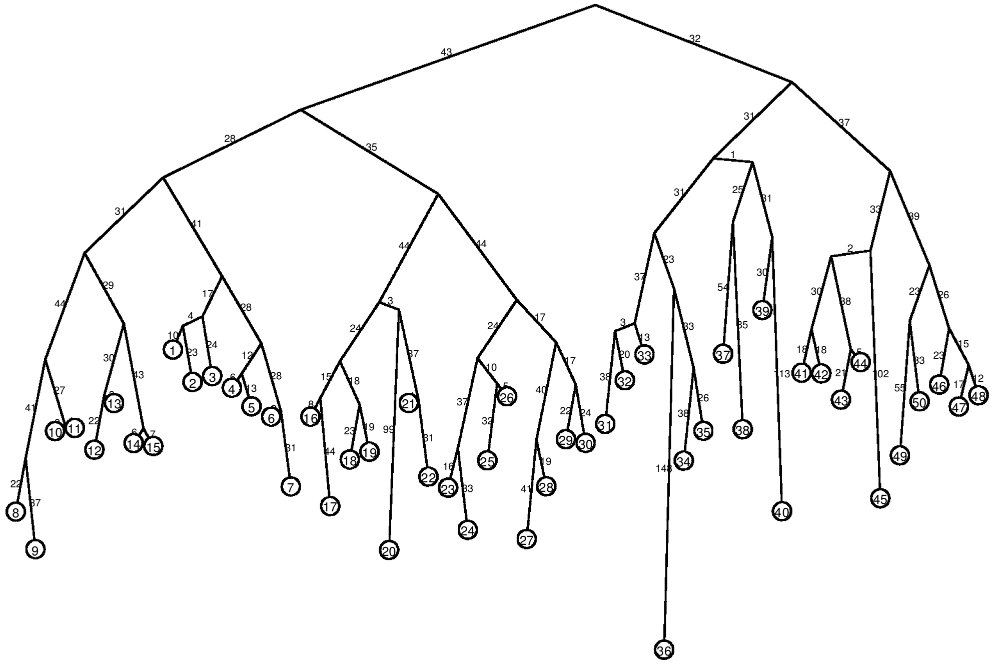


FIG. 12. Comparison of different MSA methods: the CS score (second column) is calculated using a TSP ordering. The upper bound is the CS score based on the optimal pairwise alignment.

4. DISCUSSION

We have defined a new scoring function for the evaluation of MSAs, called CS measure, that is based on a Markovian model of evolution. The frequently used SP measure, which calculates the sum of the scores of all pairwise alignments, ignores the structure of any associated phylogenetic tree and thus weights some evolutionary events more than others. This can be avoided by traversing the tree in a circular order, where all edges are traversed exactly twice. Any other noncircular tour results in a longer path, because some edges are traversed more than twice, and hence some evolutionary events are counted more often, which corresponds to a lower probability. Therefore, the shortest cycle through the tree, which is a circular tour, yields the highest CS score. Such a tour can be calculated with any TSP algorithm.

The TSP application allows us to avoid the calculation of an evolutionary tree altogether, and only the sequences at the leaves are needed with their OPA or MPA scores. In addition, the CS measure yields a direct connection between an MSA and the associated evolutionary tree, because the formula for the calculation of the score is exactly the same. The CS measure based on the OPA scores gives an upper bound on the score of the optimal MSA and evolutionary tree.

To illustrate the new scoring tool, we simulated evolution by generating random trees according to a Markovian model with the corresponding MSA. The CS score of the generated alignment was then compared to the score of alignments calculated by different methods (MSA, ClustalW, PAS, and MAP). For less than twenty sequences, MSA and PAS gave the best score, whereas for larger samples, PAS and ClustalW calculated the best alignments. A better assessment must come with actual experimental sequence data though.

With the new CS measure, we now have the possibility of improving current algorithms and finding new algorithms by maximizing the scoring function. The most simple (and expensive) approach would be a standard dynamic programming algorithm. An approximation algorithm is presented in Korostensky and Gonnet (1999a) that calculates an MSA which is within $\frac{n-1}{n} \cdot CS_{max}(S)$ for a set of sequences S (where $CS_{max}(S)$ is the upper bound). The CS scoring function can also be used for evolutionary trees, and a tree construction algorithm that is presented in Korostensky and Gonnet (1999b). The algorithms have been implemented into the Darwin system (Gonnet, 1994b) an are available via our cbrg server at: `http://cbrg.inf.ethz.ch`

REFERENCES

Altschul, S., and Erickson, B.W. 1986. Optimal sequence alignment using affine gap costs. *J. Mol. Biol.*, 48, 603.

Altschul, S., and Lipman, D. 1989. Trees, stars, and multiple sequence alignment. *J. Appl. Math.*, 49, 197–209.

Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M.A. 1994. Hidden markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA*, 91, 1059–1063.

Brenner, S.A., Cohen, M.A., and Gonnet, G.H. 1993. Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Molecular Biology*, 229: 1065–1082.

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

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1978. A model for evolutionary change in proteins. In Dayhoff, M.O., editor, *Atlas of Protein Sequence and Structure*, volume 5, 345–352.

Dress, A., and Steel, M. 1993. Convex tree realization of partitions. *Appl. Math. Lett.*, 5, 3–6.

Estabrook, G., Johnson, C., and McMorris, F. 1975. An idealized concept of the true cladistic character. *Math. Biosciences*, 23, 263–72.

Estabrook, G., Johnson, C., and McMorris, F. 1976. A mathematical foundation for the analysis of cladistic character compatibility. *Math. Biosciences*, 29, 181–87.

Felsenstein, J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Amer. J. Human Genetics*, 25, 471–492.

Felsenstein, J. 1981. Evolutionary trees from gene frequencies and quantitative characters: finding maximum likelihood estimates. *Evolution*, 35, 1229–1242.

Foulds, L.R., and Graham, R.L. 1982. The Steiner problem in phylogeny is np-complete. *Proc. Natl. Academy Science*, 3, 43–49.

Gonnet, G.H. 1994a. New algorithms for the computation of evolutionary phylogenetic trees. In Suhai, S., ed., *Computational Methods In Genome Research*, 153–161.

Gonnet, G.H. 1994b. A tutorial introduction to computational biochemistry using Darwin.

Gonnet, G.H., and Brenner, S.A. 1996. Probabilistic ancestral sequences and multiple alignments. In *Fifth Scandinavian Workshop on Algorithm Theory, Reykjevik July 1996.*

Gonnet, G.H., Cohen, M.A., and Brenner, S.A. 1992. Exhaustive matching of the entire protein sequence database. *Science*, 256, 1443–1445.

Gotoh, O. 1982. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162, 705–708.

Groetschel, M., and Holland, O. 1991. Solution of large-scale symmetric traveling salesman problems. *Math. Programming*, 141–202.

Gupta, S., Kececioglu, J., and Schaffer, A. 1995. Making the shortest-paths approach to sum-of-pairs multiple sequence alignment more space efficient in practice. *Proc. 6th Symp. On Combinatorial Pattern Matching*, 128–43.

Gupta, S.K., Kececioglu, J., and Schaffer, A.A. 1996. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *J. Comp. Biol.*

Higgins, D., and Sharp, P. 1989. Fast and sensitive multiple sequence alignments on a microcomputer. *CABIOS*, 5, 151–153.

Huang, X. 1994. On global sequence alignment. *CABIOS*, 10(3), 227–235.

Jiang, T., and Wang, L. 1994. On the complexity of multiple sequence alignment. *J. Comp. Biol.*, 1, 337–48.

Jiang, T., Wang, L., and Lawler, E.L. 1996. Approximation algorithms for tree alignment with a given phylogeny. *Algorithmica*, 16, 302–15.

Johnson, D. 1987. More approaches to the travelling salesman guide. *Nature*, 330, 525.

Johnson, D. 1990. Local optimization and the traveling salesman problem. In *Proc. 17th Colloq. On Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Sciences*, 446–461, Berlin. Springer Verlag.

Kececioglu, J. 1993. The maximum weight trace problem in multiple sequence alignment. *Proc. 4th Symp. on Combinatorial Pattern Matching*, 106–19.

Korostensky, C., and Gonnet, G.H. 1999a. Near optimal multiple sequence alignments using a traveling salesman problem approach. *SPIRE99*, 105–114.

Korostensky, C., and Gonnet, G.H. 1999b. Using traveling salesman problem algorithms for evolutionary tree construction. *J. of Comp. Biol.*, submitted.

Krogh, A., Brown, M., Mian, I.S., Sjolander, K., and Haussler, D. 1994. Hidden markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235, 1501–1531.

Lipman, D.J., Altschul, S.F., and Kececioglu, J.D. June 1989. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA*, 86, 4412–4415.

Needleman, S.B., and Wunsch, C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48, 443–453.

Padberg, M., and Rinaldi, G. 1991. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33, 60–100.

Papadimitriou, C.H. 1977. The euclidean traveling salesman problem is np-complete. *Theoretical Computer Science*, 4(3), 237–244.

Roui, R., and Kececioglu, J. 1998. Approximation algorithms for multiple sequence alignments under a fixed evolutionary tree. *Discrete Applied Mathematics*, 355–366.

Sankoff, D. 1975. Minimal mutation trees of sequences. *SIAM J. Appl. Math.*, 28, 35–42.

Sankoff, D., and Cedergren, R. 1983. Simultaneous comparison of tree or more sequences related by a tree. In Sankoff, D., and Kruskal, G., eds., *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*, volume 28, 253–263. Addison Wesley, Reading, MA.

Schwarz, R, and Dayhoff, M. 1979. Matrices for detecting distant relationship. In Dayhoff, M., ed., *Atlas of Protein Sequences*, 353–58. Natl. Biomed. Res. Found.

Smith, T.F., and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.*, 147, 195–197.

Thompson, J., Higgins, D., and Gibson, T. 1994. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22, 4673–4680.

Thorne, J., Kishino, H., and Felsenstein, J. 1993. Inching toward reality: An improved likelihood model of sequence evolution. *J. Mol. Biol.*, 34, 3–16.

Wang, L., and Gusfield, D. 1996. Improved approximation algorithms for tree alignment. *Proc. 7th Symp. on Combinatorial Pattern Matching*, 220–33.

Address correspondence to:
Chantal Korostensky
Institute for Scientific Computing
ETH Zurich
8092 Zurich, Switzerland

E-mail: Chantal.Roth@nobilitas.com