Evaluation Measures of Multiple Sequence Alignments

Gaston H. Gonnet, *Chantal Korostensky and Steve Benner
Institute for Scientific Computing
ETH Zurich, 8092 Zuerich, Switzerland
phone: ++41 1 632 74 79
e-mail: {gonnet,korosten}@inf.ethz.ch

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 structure, functionality and ultimately 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, which allows us to use the measure to evaluate evolutionary trees and 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.

Keywords: 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 familial 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.

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

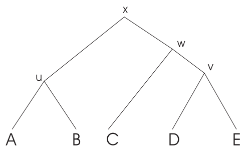


Figure 1: The sequences $\{A, B, C, D, E\}$ of the MSA on the left are related by the evolutionary tree on the right.

Example: In the MSA in Figure 1 five sequences are aligned ($n = 5$). 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 $\Sigma' = \Sigma \cup \{\text{"\_"}\} \notin \Sigma$. $\forall a_i \in \mathrm{A} : |a_i| = k$. 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 $T(S)$ 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.*

Constructing trees, MSAs, and ancestral sequences 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 Margaret Dayhoff (Dayhoff et al., 1978) and subsequently amplified, are used (Benner et al., 1993).

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; Gupta et al., 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; Estabrook et al., 1976; Felsenstein, 1973; Felsenstein, 1981; Thorne et al., 1993) are 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 1.4 *An MSA scoring function is a function* $F : \mathrm{A} \rightarrow I\!R$.

Definition 1.5 *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 many 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 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 (Altschul, 1989; Thompson et al., 1994; Higgins and Sharp, 1989) that assess the quality of an MSA, following 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.

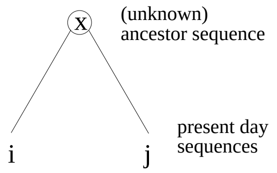
```
VNRLQQNIVSL____________EVDHKVANYKPQVEPFGHGPIFMATALVPGLYLLPL
VNRLQQSIVSLRDAFNDGTKLLEELDHRVLNYKPQANPFGNGPIFMVTAIVPGLHLLPI
```

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, 1991), 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 (Benner 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 amino acid i in one protein being aligned to amino acid j in the other is 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).



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

Definition 1.6 *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 1.7 *The optimal pairwise alignment $OPA(s_1, s_2)$ of two sequences s_1, s_2 is 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

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 2.5) are added up. SP methods are obviously deficient from an evolutionary perspective. 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

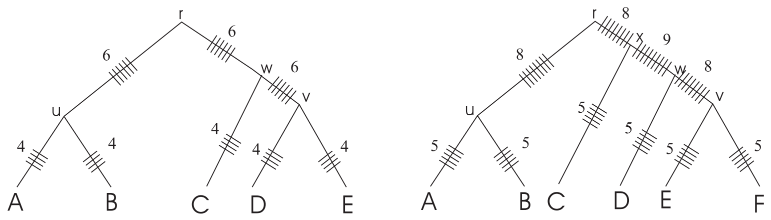


Figure 2: Traversal of a trees using the SP measure. Some edges are traversed more often than others.

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 "ticks" to the evolutionary tree 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. 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 evolutionarily perspective for scoring MSAs. This was the motivation to developed 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 an 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, 1990; Johnson, 1987), 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).

2 Methods

The task of generating an evolutionary unbiased score for a multiple sequence alignment 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 4 leaves, numbered from 1 to 4.

2.1 Circular tours

Definition 2.1 *A* <u>*Circular order*</u> $C(T)$ *of 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.

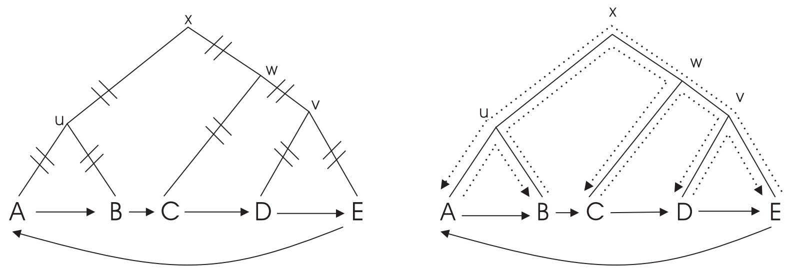


Figure 3: Traversal of a tree in circular order

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).*

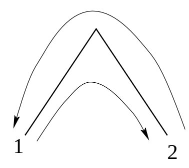


Figure 4: circular tour for 2 leaves

When we have a tree with n leaves, subdivide the tree anywhere into the subtree1 and subtree2 (see Figure 5) and look at the edge in the middle that connects the two subtrees. When the tree is traversed in the order $\langle 1, .., i, .., j, .., n \rangle$ *and back to* 1 *in a circular tour, the edge in the middle is traversed 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 leaf.*

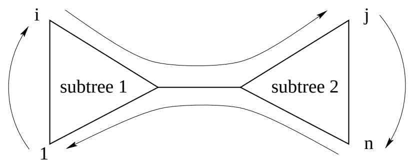


Figure 5: circular tour for n leaves

Lemma 2.2 (Non-circular Tours) *Any other non-circular tour traverses at least one edge more than twice.*

Proof 2.2 *Again take a tree with* n *leaves, and subdivide the tree anywhere into the subtree1 and subtree2 (see Figure 6). When the tree is traversed in the order* $\langle 1, .., i, .., j, .., n\rangle$ *and back to* 1*, but this time the leaves* i *and* j *are swapped, the edge in the middle is traversed four times.*

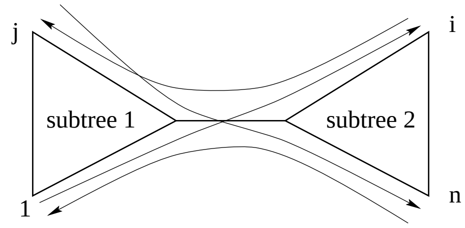


Figure 6: Non-circular tour for n leaves

For each tree T there exist $2^{(n-2)}$ isomorphic forms. An isomorphic tree T' has the same tree topology as T, and it differs from T only in the sense that a subtree is rotated (in a graphical representation).

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 one of the subtrees (leaves* j *and* n *are now on the opposite side as before)(see Figure 7). When tree tree is traversed again in the order* $\langle 1, .., i, .., j, .., n\rangle$ *and back to* 1*, the middle edge is only traversed twice. Again, since the division into subtrees can be done anywhere in the tree, all edges are counted twice, and this is true for all isomorphic trees.*

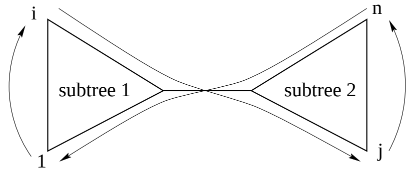


Figure 7: circular tour of an isomorphic tree

An immediate conclusion from this is that any tree has $2^{(n-2)}$ different circular tours which correspond to the circular orders of the isomorphic trees.

Example: The order $\langle A, B, C, D, E\rangle$ is a circular order in Figure 8, 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.

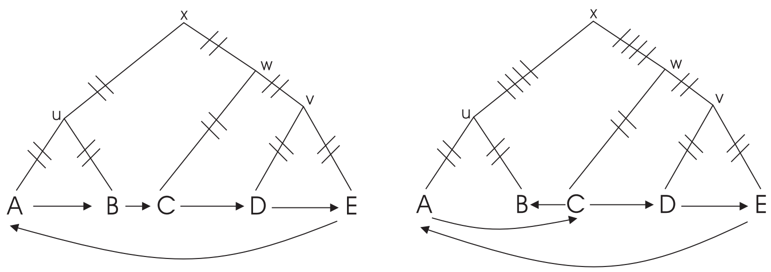


Figure 8: Traversal of a phylogenetic tree in a circular $\langle A, B, C, D, E\rangle$ and non-circular order $\langle A, C, B, D, E\rangle$

2.2 Traveling Salesman Problem Application

The probability (exponential of the score) 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 scores by aligning each sequence with every other sequence using a DP algorithm to obtain the Optimal Pairwise Alignment. We shall refer to these as OPA scores (Carillo and Lipman, 1988), to distinguish (see below) these from a pairwise alignment inferred from an MSA. These scores can be obtained without either a tree or an MSA. Our goal is to be able to find an circular tour without the need of constructing an evolutionary tree.

The score of a pairwise alignment is the logarithm of the probability of the evolution happening, as we discussed above. In Figure 9 the score of sequences B aligned against sequence C is shown. The score of the pairwise alignment corresponds to evaluating parts of the evolution from x to B and from x to C.

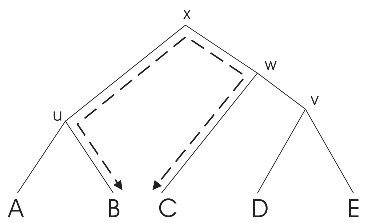


Figure 9: Score of a pairwise alignment

Since the scores are logarithms of probabilities, the sum of the OPA scores in circular order divided by two corresponds to the probability of the entire evolution happening for the given set of sequences.

As we have shown above, a circular tour is the shortest possible tour for a tree. A shorter distance corresponds to a higher probability. So when we talk about shorter tours, we also mean higher score. A non-circular tour has at least some edge of the tree traversed more than twice, and no edge less than twice. So at least one evolutionary event is counted more than twice and hence the probability would be lower.

We are only interested in the most likely tree, that is the tree with the maximum score, or maximum probability. In this paper we have so far always been using some tree, which we don't have in reality. The only information available to us is just the sequences and the OPA scores.

So suppose now we do not have any information about the best tree for that given set of sequences. But we know that the best tree has some circular tour. We also know that a circular tour is the shortest possible tour through that tree, and that the best tree has the shortest total path length (sum of all edges) of all possible trees, since we want the tree with the maximum probability (see also section 3.1).

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

Definition 2.2 *The TSP order $TSP(S)$ 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, 1990; Johnson, 1987). The problem is to find the shortest tour where each city is visited once. We use a modified version of the problem: in our case, the cities correspond to the sequences and the distances are the scores of the pairwise alignments. In addition, we are interested in the *longest*, not shortest tour, as our scores represent probabilities and we are interested in the *maximum* probability [1]. The TSP 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 have seldom 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 scores.

We refer to the sum of OPA scores in circular order $C(S)$ as $CS_{max}(S)$.

Definition 2.3 *Let $C(S)$ be the output permutation of a TSP algorithm, then $CS_{max}(S)$ for a given set of sequences $S = \{s_1, s_2, .., s_n\}$ is:*
$CS_{max}(S) = \frac{1}{2} \sum_{i=1}^{n} OPA(s_{C_i}, s_{C_{i+1}})$, *where* $C_{n+1} = C_1$.

2.3 Scoring of MSAs without evolutionary trees

The CS algorithm provides a method to evaluate a specific MSA or to compare different algorithms for constructing MSAs. When an MSA is scored using the CS measure, the score is the probability of such an alignment happening due to common ancestry compared to random sequences. To do so, we abandon scores obtained pairwise from OPAs. Instead, the pairwise relationship between two sequences in the protein family are extracted from the MSA itself. These are scored using a Dayhoff matrix without dynamic programming re-alignment to give an optimal alignment. These are MSA-derived pairwise alignments (MPA).

[1] To use any available TSP algorithm, we simply subtract the scores from a large number and then compute the shortest tour.

Definition 2.4 *The function* $\underline{S(x,y)}$ *scores two symbols* $x, y \in \Sigma'$*:*

$$S(x,y) = \begin{cases} DM(x,y),\ \textit{if } x \neq \text{"\_"} \textit{and} y \neq \text{"\_"} \\ \textit{0 if } x = \text{"\_"} \textit{and} y = \text{"\_"} \\ \textit{otherwise an affine gap cost that depends on the gap length (Benner et al., 1993)} \end{cases}$$

DM is an entry in a Dayhoff matrix (Gonnet et al., 1992).

The function $S(x,y)$ (see Definition 2.4) scores two symbols x, y that are either amino acids or gap characters in our case.

Definition 2.5 *The induced MSA-derived pairwise alignment* $\underline{MPA(a_i, a_i)}$ *of two sequences* $a_i, a_j \in \mathrm{A}$ *is:*

$$MPA(a_1, a_2) = \sum_{m=1}^{k} S(a_i[m], a_j[m])$$

MPA scores must be equal to or less than the scores obtained from the OPA pairwise alignments. The CS method can be applied now again without the need for an explicitly defined evolutionary tree by repeating the procedure outlined above, but using MPA scores instead of OPA scores. The CS score of the MSA is the sum of MPA scores in the circular order C divided by two:

Definition 2.6 *The score* $\underline{CS(\mathrm{A})}$ *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* $C_{n+1} = C_1$*.*

Note that 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 [2].

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 *OPA scores are optimal. Therefore,* $CS_{max}(S)$*, which is the sum of OPA scores in circular order obtained by a TSP algorithm, is also maximal. Assume this last statement is wrong there exists another score* $CS'(\mathrm{A})$ *that is higher than* $CS_{max}(S)$*. Since that score* $CS'(\mathrm{A})$ *is a sum of MPA scores, it would follow that at least one of the MPA scores is higher than the OPA score, which is a contradiction.*

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.

Note that when none of the optimal pairwise alignments of S have gaps, then the the score of the optimal MSA $CS(\mathrm{A}) = CS_{max}(S)$.

[2] The gaps are scored according to the formula $a + l \times b$, 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 that also depends on the PAM distance (Benner et al., 1993)

3 Connection between Trees and Msas

We now show that the upper bound $CS_{max}(S)$ also serves as an upper bound for the score of the associated, unknown evolutionary tree $T(S)$. In addition, the scoring function $CS(\mathrm{A})$ serves as a scoring function for a tree $T(\mathrm{A})$.

3.1 Probability of an evolutionary tree configuration

To obtain the probability for an entire evolutionary tree configuration, one must sum the logarithms of the probabilities of each event represented by the tree, which yields an overall score that is equivalent to the logarithm of the product of the probabilities of each event. For the example tree in Figure 10, the total evolutionary history has six associated probabilities: A→B, A→C, B→1, B→2, C→3 and C→4.

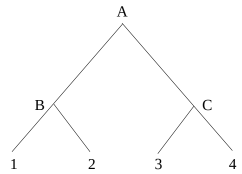


Figure 10: A sample evolutionary tree

Definition 3.1 *The* <u>*tree*</u> $T(\mathrm{A})$ *associated with an MSA* A *is the tree that corresponds to the evolutionary history of the MSA* A.

Definition 3.2 *An* <u>*tree scoring function*</u> *is a function* $F : T \to \mathbb{R}$.

If we knew the sequences of the ancestral proteins at the internal nodes of the tree, the probability of the entire tree is:

$$Pr\{\text{tree of Figure 10}\} = Pr\{\text{ A}\to\text{B }\}Pr\{\text{ A}\to\text{C }\}Pr\{\text{ B}\to\text{1 }\}Pr\{\text{ B}\to\text{2 }\}Pr\{\text{ C}\to\text{3 }\}Pr\{\text{ C}\to\text{4 }\} \tag{4}$$

where $Pr\{\text{ X}\to\text{Y }\}$ is interpreted as the probability of mutating from the amino acid in the node X to the amino acid in node Y according to the distance of the edge X-Y. The individual probabilities would be obtained by aligning the protein sequences at the beginning and end of each episode of evolution and scoring them using a Dayhoff matrix (for example) as described above.

Normally, of course, the sequences for the ancestor proteins A, B and C are not known, as the organisms that contained them have long since died. In (Gonnet and Benner, 1996), a formula for the probability of an entire evolutionary configuration (the ensemble of the phylogenetic tree and the corresponding MSA) is derived. It corresponds exactly to the notion of computing the probability of traversing each edge of the tree. We can compute the probability of the tree configuration by adding the probabilities of all episodes represented by each of the edges.

Hence a tree scoring function $F(T)$ must score each edge of the tree exactly once.

Lemma 3.1 *Let $T(A)$ be the the optimal tree for the MSA* A *(the tree with the maximum probability). The function $CS(\mathrm{A})$ corresponds to the probability of T.*

Proof 3.1 *The function $CS(\mathrm{A})$ scores each edge in $T(\mathrm{A})$ exactly once, therefore $CS(\mathrm{A})$ can be used to score $T(\mathrm{A})$.*

Lemma 3.2 (Maximum Probability) *The maximal score $CS_{max}(S)$ is both an upper bound on the probability for the maximum probability tree and it is also an upper bound on the probability of an MSA for the underlying sequences.*

Proof 3.2 *We have already shown that $CS_{max}(S)$ is the upper bound for $CS(\mathrm{A})$ for any A derived from S. Since the formula for deriving the score is exactly the same for both trees and MSAs, $CS_{max}(S)$ also serves as an upper bound for the score of any tree $T(S)$ that can be derived from S. Therefore, there can be no tree and also no MSA with a higher probability than $CS_{max}(S)$ for a given set of sequences S.*

When we score an MSA with the CS method using MPA scores, we immediately also derive the score of the entire evolutionary configuration represented by the tree that corresponding to the MSA. So the score corresponds both to the probability that the sequences in the MSA are related, and it also corresponds to the probability that the evolution happened according to the associated tree.

We have now a way to determine a circular order and thus an upper bound on the score of an MSA and its associated evolutionary tree 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 OPA/MPA scores with each other are needed. Nevertheless the $CS(\mathrm{A})$ score corresponds to the probability of an evolutionary tree configuration.

4 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 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 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 Benner, 1996; Gonnet, 1994a), and 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 12-14).

4.1 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:

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

1       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGAKLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGAKLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGSVYLR___HVEIDLRTQVGSSPYAATVIKNVIKNTRPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGSVILK___HLEIDFQSSVGSNPRAATYVKNVIKGRKPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGSVLLR___TLELDFQGQVGTPPRAAVYVKNVTKGAKPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGSVVLR___HFELDFEGQAETNPRSSVYVKNVSQGVEPIHLTEWQEFNYGNVSCKI
3       LDVLDVTT___________IIIQ___TFRIDLQEQLGSNPASATYVKNILTGAKLLHLSEGEEYTMGHAVLIM
4       LDVLDVVT___________IIVV___TFRIDLQEQVGENPASASYVQDILTGAKLLHLSDGKEYTMGHVVAII
```

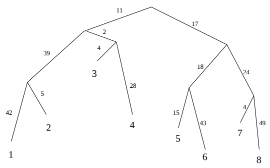


Figure 11: Tree associated with example MSA

The score of the alignment is the MPA score introduced above. 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 11. The numbers on the edges of the tree are PAM distances.

The maximum possible score, the $CS_{max}(S)$ score (see Definition 2.3), 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 2.5), 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. This is an indication that the correct circular order was found.

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

```
Sequences:
----------
1 LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGAKLLHISDGQDTTIGQTVAIT
2 LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGAKLLHLSDGQESAMGQVVAIT
3 LDVLDVTTIIIQTFRIDLQEQLGSNPASATYVKNILTGAKLLHLSEGEEYTMGHAVLIM
4 LDVLDVVTIIVVTFRIDLQEQVGENPASASYVQDILTGAKLLHLSDGKEYTMGHVVAII
5 LHVIDERKRLPAARFNKGSVILKHLEIDFQSSVGSNPRAATYVKNVIKGRKPLKLCDGQEISLGLIVCIW
6 LHVIDERKRLEAARFNKGSVYLRHVEIDLRTQVGSSPYAATVIKNVIKNTRPLKLCMGQELSLGMIVMLF
7 LAVIEVRRGQVALEFNKGSVLLRTLELDFQGQVGTPPRAAVYVKNVTKGAKPLHLVEGQEFNLGYVTCII
8 VHVADVTRGLTRLEFDKGSVVLRHFELDFEGQAETNPRSSVYVKNVSQGVEPIHLTEWQEFNYGNVSCKI

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       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGAKLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGAKLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGSVYLR___HVEIDLRTQVGSSPYAATVIKNVIKNTRPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGSVILK___HLEIDFQSSVGSNPRAATYVKNVIKGRKPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGSVLLR___TLELDFQGQVGTPPRAAVYVKNVTKGAKPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGSVVLR___HFELDFEGQAETNPRSSVYVKNVSQGVEPIHLTEWQEFNYGNVSCKI
3       LDVLDV___________TTIIIQ___TFRIDLQEQLGSNPASATYVKNILTGAKLLHLSEGEEYTMGHAVLIM
4       LDVLDV___________VTIIVV___TFRIDLQEQVGENPASASYVQDILTGAKLLHLSDGKEYTMGHVVAII
```

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 − the CS score were 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       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGAKLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGAKLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGS___VYLRHVEIDLRTQVGSSPYAATVIKNVIKNTRPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGS___VILKHLEIDFQSSVGSNPRAATYVKNVIKGRKPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGS___VLLRTLELDFQGQVGTPPRAAVYVKNVTKGAKPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGS___VVLRHFELDFEGQAETNPRSSVYVKNVSQGVEPIHLTEWQEFNYGNVSCKI
3       LDVLDVTT______________IIIQTFRIDLQEQLGSNPASATYVKNILTGAKLLHLSEGEEYTMGHAVLIM
4       LDVLDVVT______________IIVVTFRIDLQEQVGENPASASYVQDILTGAKLLHLSDGKEYTMGHVVAII
```

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       LETIDICKGCAALEYYRGPMIMRAMTSFRLDIKQQVGTTKACADATSNELTGAKLLHISDGQDTTIGQTVAIT
2       LELIDIEKGCAALEYNKGSMIMRAMTTFRLDLKDQVGSTAACADATKNKLTGAKLLHLSDGQESAMGQVVAIT
6       LHVIDERKRLEAARFNKGSVYLR___HVEIDLRTQVGSSPYAATVIKNVIKNTRPLKLCMGQELSLGMIVMLF
5       LHVIDERKRLPAARFNKGSVILK___HLEIDFQSSVGSNPRAATYVKNVIKGRKPLKLCDGQEISLGLIVCIW
7       LAVIEVRRGQVALEFNKGSVLLR___TLELDFQGQVGTPPRAAVYVKNVTKGAKPLHLVEGQEFNLGYVTCII
8       VHVADVTRGLTRLEFDKGSVVLR___HFELDFEGQAETNPRSSVYVKNVSQGVEPIHLTEWQEFNYGNVSCKI
3       LDVLDVTT___________III___QTFRIDLQEQLGSNPASATYVKNILTGAKLLHLSEGEEYTMGHAVLIM
4       LDVLDVVT___________IIV___VTFRIDLQEQVGENPASASYVQDILTGAKLLHLSDGKEYTMGHVVAII
```

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 underlying tree structure.

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

Figure 12 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). In this and all the following examples the tables show the data that was collected from over a hundred MSAs each.

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

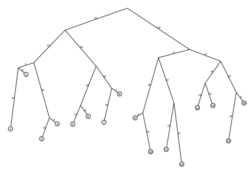


Figure 12: Comparison of different MSA methods: the CS score (second column) is calculated using a TSP ordering. The upper bound is the CS score based on the optimal pairwise alignment.

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 on the right is an example of a generated phylogenetic tree.

The next trees (Figure 13) 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 14) 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.

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

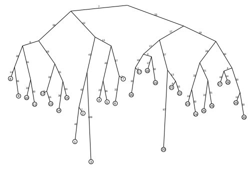


Figure 13: Comparison of different MSA methods: the CS score (second column) is calculated using a TSP ordering. The upper bound is the CS score based on the optimal pairwise alignment.

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

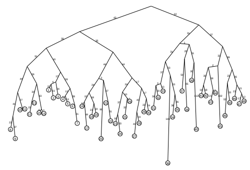


Figure 14: Comparison of different MSA methods: the CS score (second column) is calculated using a TSP ordering. The upper bound is the CS score based on the optimal pairwise alignment.

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 $CS(\mathrm{A})$ of the simulated MSA. Obviously the ultimate evaluation of tools must be done with real rather than simulated data.

5 Discussion

We have defined a new scoring function, called CS measure, for the evaluation of MSAs 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 non-circular 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 an 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. This allows us to use the CS measure evaluate evolutionary trees. 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 have now the possibility of improving current algorithms and finding new heuristics by maximizing the scoring function. The description of the new methodologies exceeds the scope of this paper and will be available in (Korostensky and Gonnet, 1999).

References

Altschul, S. (1991). Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.*, 219:555 – 65.

Altschul, S. F. (1989). Gap costs for multiple sequence alignment. *J. theoretical Biology*, 138:297–309.

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.

Benner, S. A., Cohen, M. A., and Gonnet, G. H. (1993). Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. 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, pages 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., editor, *Computational Methods In Genome Research*, pages 153–161.

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

Gonnet, G. H. and Benner, 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 Benner, 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 salesmand problems. *Math. Programming*, pages 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*, pages 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. In *J. Computational Biology*.

Higgins, D. and Sharp, P. (1989). Fast abd 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 Science*, pages 446 – 461, Berlin. Springer Verlag.

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

Korostensky, C. and Gonnet, G. H. (1999). Near optimal multiple sequence alignments using a traveling salesman problem approach. *SPIRE99*. 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. Molecular Biology*, 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.

Roui, R. and Kececioglu, J. (1998). Approximation algorithms for multiple sequence alignments under a fixed evolutionary tree. *Discrete Applied Mathematics*, pages 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., editors, *Time Warps, String Edits, and Marcomolecules: the Theory and Practice of Seqeunce Comparison*, volume 28, pages 253 – 263. Addison Wesley, Reading MA.

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. Molecular Biology*, 34:3–16.

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