A Simplified Proof of the NP- and MAX SNP-hardness of Multiple Sequence Tree Alignment

H. Todd Wareham
Computer Science Department
University of Victoria
Victoria, British Columbia V8W 3P6, Canada
E-mail: harold@csr.uvic.ca

August 3, 1995

This communication gives a short proof of the NP- and MAX-SNP hardness of the multiple sequence tree alignment problem. As such, it gives the first available proof of the NP-hardness result claimed in (Warnow 1993) and greatly simplifies the MAX SNP-hardness proof given in (Wang and Jiang 1994). Though there are many types of multiple sequence alignment (see (Chan *et al.* 1994) and references), tree alignment is preferable because it takes evolutionary descent into account in creating the alignment (Altschul and Lipman 1989; Sankoff and Cedergren 1983). The best known algorithms for this problem require time exponential in the length of the longest given sequence (see (Jiang *et al.* 1994) and references). The NP- and MAX SNP-hardness results below are important because they imply that, unless $P = NP$, there are no polynomial-time algorithms for deriving exact solutions or polynomial-time approximation schemes which guarantee approximate solutions of cost arbitrarily close to optimal for the multiple sequence tree alignment problem (see (Garey and Johnson 1979; Johnson 1992) for details).

Given a set of strings $X = \{x_1, \ldots, x_k\}$ on an alphabet Σ, an *alignment* of X is a set of strings $A = \{a_1, \ldots, a_k\}$, $|a_1| = |a_2| = \ldots = |a_k| = n$, on augmented alphabet $\Gamma = \Sigma \cup \{\Delta\}$ such that each string a_i is a copy of x_i into which $n - |x_i|$ copies of special symbol Δ have been inserted (Kececioglu 1993; Pevzner 1992). Symbol Δ is called an *indel* and represents the insertion or deletion of a particular symbol in one string relative to another. An alignment can be visualized as a $k \times n$ matrix, in which row i corresponds to string i and column j is the vector of symbols at position j over all given strings. A *gap* is a maximal string of indels within a row of an alignment. We will assume that no alignment in this paper has one or more columns composed purely of indels. The cost of an alignment is generally computed relative to a given $\Gamma \times \Gamma$ symmetric matrix M, where $M(x, y)$, $x, y \in \Gamma$, is the cost of converting symbol x to symbol y.

Define a *tree-alignment* T *of* X as a tree $T = (V, E)$ such that each vertex is labeled with a different string from Γ^p, $p \geq \max_{x_i \in X} |x_i|$, and each string $x \in X$ corresponds to a unique vertex-label string $v \in V$ that is a copy of x into which $p - |x|$ indels have been inserted. Let $l(v)$, $v \in V$, be the vertex-label string associated with vertex v in T, and $l(v)_i$ be the i-th character in that vertex-label string. Implicit in T is an $|V| \times p$ matrix which corresponds to an alignment of all vertex-label strings in $|V|$; denote this alignment by T_A. Note that T_A contains a submatrix corresponding to an alignment of all strings in X. Following (Altschul 1989), define a gap in a tree alignment as a maximal sequence of indels in a row r of T_A that is matched against non-indels in at least one of the rows of T_A corresponding to the vertices adjacent to the vertex corresponding to r in T. The cost of tree alignment T relative to cost-matrix M is then

$$c_{TA}(T, M) = \sum_{i=1}^{p} \sum_{(x,y) \in E} M(l(x)_i, l(y)_i)$$

This yields the following problem:

MULTIPLE SEQUENCE TREE ALIGNMENT (MSTA)
Instance: A set X of k strings on alphabet Σ, a $\Gamma \times \Gamma$ symmetric matrix M, and an integer m.
Question: Is there a tree-alignment T of X such that $c_{TA}(T, M) \leq m$?

Problem MSTA was claimed to be NP-hard in (Warnow 1993), and was shown to be MAX SNP-hard in (Wang and Jiang 1994).

Consider the following alternate proof that yields both NP- and MAX SNP-hardness for MSTA via a reduction from the UNWEIGHTED BINARY WAGNER PHYLOGENY problem. Let H_2^d denote the d-dimensional binary hypercube graph, i.e., $H_2^d = (V, E)$ where $V = \{0, 1\}^d$, the set of all strings of length d on alphabet $\{0, 1\}$, and $E = \{\{u, v\} |$ the strings for u and v differ in exactly one position$\}$. Given a tree T in H_2^d, let the *length* of T be the number of edges in T.

UNWEIGHTED BINARY WAGNER PHYLOGENY (UBW)
Instance: A positive integer d, a subset S, $|S| = n$, of $\{0, 1\}^d$, and a positive integer B.
Question: Is there a tree in H_2^d that includes S and whose length is $\leq B$?

Problem UBW is NP-complete by (Day *et al.* 1986; Graham and Foulds 1982), and was shown to be MAX SNP-hard by the former reduction in (Wareham 1993). The biological interpretation of the problem is that each dimension of the binary hypercube corresponds to an unordered two-state character, each member of S is a species defined by its values for each of the d characters, and each edge $\{u, v\}$ in the binary hypercube corresponds to an evolutionary change, i.e., $0 \rightarrow 1$ or $1 \rightarrow 0$, in a single character for the species corresponding to vertices u and v. As each edge encodes a single character-state change, the tree on S that has the minimum number of edges accounts for the distribution of characteristics of the species in S by the fewest total number of evolutionary changes, and is hence preferred under the criterion of phylogenetic parsimony (see (Wareham 1993), Chapter 3.1 and references).

Our proof uses the following convenient property.

Lemma 1 *For any instance of UBW, there is a trivial solution tree whose length is* $\leq (n-1)d$.

Proof: Any two vertices u and v in H_2^d may be connected by a path of at most d edges, each of which corresponds to one of the d possible character differences between the strings for u and v. Given any ordering of S, simply join each vertex to its successor in the ordering by such a path. This collection of paths may intersect, causing cycles in the resulting graph G; however, it is a fairly simple matter to compute the minimal spanning tree of this graph by a standard algorithm and then delete all paths leading to leaves that are not members of S. The produced tree T' is still a tree in H_2^d, still contains all vertices in S, and cannot have more edges than the original graph G. As G had $l \leq (n-1)d$ edges, T' has length $\leq l \leq (n-1)d$, completing the proof. □

We can assume by this lemma that we are only interested in solution trees of length $< (n-1)d$.

Theorem 1 *MSTA is NP-complete.*

Proof: As any solution tree alignment matrix for an instance I of MSTA has at most $2k-1$ rows and kl columns, where l is the length of the longest string in X, and the cost associated with that alignment can be computed in time polynomial in the size of I, MSTA is in NP. To show that MSTA is NP-hard, we will reduce UBW to MSTA. Given an instance of UBW, construct the following instance of MSTA: Let $\Sigma = \{0, 1\}$, $X = S$, $k = n$, $m = B$, and M be the cost matrix in Table 1.

	0	1	Δ
0	0	1	$(n-1)d$
1	1	0	$(n-1)d$
Δ	$(n-1)d$	$(n-1)d$	0

Table 1: Cost Matrix M for MSTA NP-hardness Reduction.

This instance can be constructed in time polynomial in the size of the given instance of UBW. Note that for $u, v \in \{0, 1\}^d$, the cost c for edge $\{u, v\}$ under M is simply the Hamming distance between u and v; hence, by the arguments in Lemma 1, such an edge may be replaced by an appropriate path in H_2^d composed of c edges.

We now show that an instance of UBW has a solution tree of length at most B if and only if the constructed instance of MSTA has a tree alignment of cost at most B.

UBW implies MSTA: It is obvious that any tree on S in H_2^d of length B corresponds directly to a tree alignment of X of cost B under M.

MSTA implies UBW: We first show that each sequence in any solution tree alignment for an instance of MSTA constructed as above is a vertex-label for some vertex in H_2^d. Suppose there is a solution tree alignment T_A that has a column which contains both indels and one or both of 1's and 0's. Such a column adds at least $(n-1)d$ to the cost of T_A. However, by Lemma 1, we can assume that all solution trees (and hence solution tree alignments) have cost at most $(n-1)d$. Hence, no solution tree alignment can contain such a column and have the required cost m. Therefore, solution tree alignments can contain only columns composed of 1's and 0's.

As each vertex in the solution tree alignment directly corresponds to the vertex-label of some vertex in H_2^d, by the observation made at the beginning of this proof, each edge in this alignment tree can be replaced by any of the shortest paths between the vertices in H_2^d corresponding to its endpoints. This collection of paths may intersect, causing the resulting graph G to be acyclic; however, by the arguments in Lemma 1, if G has B edges, we can easily produce a tree T' of length $\leq B$, completing the proof. □

This proof establishes that MSTA is NP-complete in the very restricted case that $|\Sigma| = 2$ and the cost matrix M is ultrametric. That is, cost-matrix M on alphabet $\Gamma = \Sigma \cup \{\Delta\}$ satisfies the following five properties:

1. $\forall x \in \Gamma$, $M(x, x) = 0$.
2. $\forall x, y \in \Gamma$, $M(x, y) = 0$ implies $x = y$.
3. $\forall x, y \in \Gamma$, $M(x, y) = M(y, x)$.
4. $\forall x, y, z \in \Gamma$, $M(x, y) \leq M(x, z) + M(z, y)$.
5. $\forall x, y, z \in \Gamma$, $M(x, y) \leq \max[M(x, z), M(z, y)]$.

A cost-matrix that satisfies conditions (1) – (3) is a *semi-metric*; if condition (4) is also satisfied, the matrix is a *metric*. A metric that also satisfies condition (5) is an *ultrametric*. Any ultrametric is thus also a metric and a semi-metric. Though metricity has on occasion been considered a requirement for cost matrices, it is ultimately only a desirable mathematical property which may not be relevant to the underlying biology (see (Fitch 1993; Wheeler 1993) and references). The significance of the NP-completeness of this restricted case is that it trivially proves NP-completeness for the more general cases typically encountered in practice, in which $|\Sigma| > 2$ or the cost matrix is only a semi-metric.

The reduction above is unrealistic because it proposes an indel cost that is dependent on n and d. We can reduce the indel cost from $(n-1)d$ to some constant c, $c > 2$, if we adopt a more biologically relevant gap weighting that adds a "start-up" cost for each gap in the alignment – in particular, we add for each gap (regardless of length) the cost $(((n-1)d)/2) - c$. The reader can verify that under this cost function, the cost of adding the minimum number of gaps to T_A, i.e., a single indel somewhere in each string of the alignment such that all indels are not in the same column, will be at least $(n-1)d$, allowing the logic of Theorem 1 to be applied again (under a new cost matrix that is still an ultrametric). Such a gap-cost scheme which assigns constant weight to each gap is known as *affine gap cost*, and is the simplest and most commonly used of a variety of gap-weighting schemes (see (Altschul 1989) and references). As affine gap cost schemes are restricted cases of many more complex and biologically useful gap cost schemes, the reduction given above also shows that MSTA is NP-complete under these other schemes.

The reduction in Theorem 1 also suffices to show the MAX SNP-hardness of MSTA. To show MAX SNP-hardness, we need the following more sophisticated form of reducibility between optimization rather than decision problems.

Definition 1 (Papadimitriou and Yannakakis 1991) *Let Π and Π' be two optimization (maximization or minimization) problems. We say that Π* linearly reduces (L-reduces) *to Π' if there are two polynomial-time algorithms f, g and constants $\alpha, \beta > 0$ such that for each instance I of Π:*

(L1) *Algorithm f produces an instance $I' = f(I)$ of Π', such that the optima of I and I', $OPT(I)$ and $OPT(I')$, respectively, satisfy $OPT(I') \leq \alpha OPT(I)$*

(L2) *Given any solution of I' with cost c', algorithm g produces a solution of I with cost c such that $|c - OPT(I)| \leq \beta|c' - OPT(I')|$.*

Theorem 2 *MSTA is MAX SNP-hard.*

Proof: The arguments for the two implications in Theorem 1 show that a solution to an instance of UBW can be easily transformed into an solution of the same cost to the constructed instance of MSTA, and that a solution to a constructed instance of MSTA can be easily transformed into a solution of equal or lesser cost to the corresponding instance of UBW. Hence, these arguments encode the required algorithms f and g in the definition above, showing that UBW L-reduces to MSTA with $\alpha = \beta = 1$. To complete the proof, note that the reduction from VERTEX COVER to UBW given in (Day *et al.* 1986) for NP-hardness also shows that UBW is MAX SNP-hard (see (Wareham 1993), Section 5.3). □

This result gains further interest in light of the following. Given a problem X and an instance I of X, let $OPT_X(I)$ be the cost of an optimal solution for I and $A_X(I)$ be the cost of the solution found for I by algorithm A for X. Problem X has a *relative approximation* c, $c > 0$, if there is a polynomial-time algorithm A such that for all instances I of X, $|OPT_X(I) - A_X(I)| \leq c \cdot OPT_X(I)$ (Garey and Johnson 1979).

Lemma 2 (Papadimitriou and Yannakakis 1991, Proposition 3) *If Π L-reduces to Π' with constants α and β, and Π' has a relative approximation ϵ, then Π has a relative approximation $\alpha\beta\epsilon$.*

Corollary 1 *If Π L-reduces to Π' with constants α and β, and "Π has a relative approximation c" implies* **X**, *then "Π' has a relative approximation $\frac{c}{(\alpha\beta)}$" implies* **X**.

By these results, the L-reduction from UBW to MSTA given above ties the relative approximability of MSTA to the relative approximability of UBW in two ways. First, this reduction establishes upper bounds for relative approximations for UBW by showing how relative approximations for MSTA can be applied directly to UBW. For example, as MSTA has a relative approximation of $\approx .747$ by a variant of the algorithm in (Berman and Ramaiyer 1994) (Wang and Jiang 1994; see also Gusfield 1993), so does UBW; this betters the previous best known result of 1 derived via the techniques in (Kou *et al.* 1981) (Wareham 1993). Second, this reduction establishes lower bounds for relative approximations for MSTA by showing how certain such approximations would induce improbable relative approximations for UBW.

Note Added In Proof: Since this paper was submitted, the manuscript cited in (Warnow, 1993) has appeared (Sweedyk and Warnow, 1995). Contrary to previous citations (Wang and Jiang, 1994; Jiang *et al.*, 1994), the problem shown to be NP-complete in their paper is not MSTA as defined here, but rather Multiple Sequence Tree Alignment With a Given Topology, when the topology is a star. Note that at present, there is no reduction from either of these problems to the other; hence, neither of the results in this paper and in (Sweedyk and Warnow, 1995) imply the other. As such, the result in (Sweedyk and Warnow, 1995) answers an open question posed in (Wang and Jiang, 1994).

Acknowledgements

The author would like to thank T. Jiang and L. Wang for sharing preprints and for various helpful e-mail conversations, the two anonymous reviewers for their very useful comments, D. Kingsbury for invaluable editorial advice and assistance, and P. Evans, M. Fellows, M. Hallett, V. King, and B. Shea for encouragement and sound advice.

References

Altschul, S.F. 1989. Gap costs for multiple sequence alignment. *J. Theor. Biol.* 138, 297–309.

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

Berman, P., and Ramaiyer, V. 1994. Improved approximations for the Steiner tree problem. *J. Algorithms* 17(3), 381–408.

Chan, S.C., Wong, A.K.C., and Chiu, D.K.T. 1992. A survey of multiple sequence comparison methods. *Bull. Math. Biol.* 54(4), 563–598.

Day, W.H.E., Johnson, D.S., and Sankoff, D. 1986. The computational complexity of inferring rooted phylogenies by parsimony. *Math. Biosci.* 81, 33–42.

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

Garey, M.R., and Johnson, D.S. 1979. *Computers and intractability.* W. H. Freeman, San Francisco.

Graham, R.L., and Foulds, L.R. 1982. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. *Math. Biosci.* 60, 133–142.

Gusfield, D. 1993. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bull. Math. Biol.* 55(1), 141–154.

Jiang, T., Lawler, E.L., and Wang, L. 1994. Aligning sequences via an evolutionary tree: complexity and approximations (extended abstract). 760–769. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing.* ACM Press, New York.

Johnson, D.S. 1992. The NP-completeness column: An ongoing guide (23rd edition). *J. Algorithms* 13, 502–524.

Kececioglu, J.D. 1993. The maximum weight trace problem in multiple sequence alignment. 106–119. In *Proceedings of the Fourth Annual Symposium on Combinatorial Pattern Matching (CPM)*. Lecture Notes in Computer Science no. 684. Springer-Verlag, Berlin.

Kou, L., Markowsky, G., and Berman, L. 1981. A fast algorithm for Steiner trees. *Acta Inform.* 15, 141–145.

Papadimitriou, C.H., and Yannakakis, M. 1991. Optimization, approximation, and complexity classes. *JCSS* 43, 425–440.

Pevzner, P.A. 1992. Multiple alignment, communication cost, and graph matching. *SIAM J. Appl. Math.* 52(6), 1763–1779.

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

Sweedyk, E., and Warnow, T. 1994. The Tree Alignment Problem is NP-Complete. Manuscript.

Warnow, T. 1993. Personal communication.

Wang, L., and Jiang, T. 1994. On the complexity of multiple sequence alignment. *J. Comp. Bio.* 1(4), 337–348.

Wareham, H.T. 1993. On the computational complexity of inferring evolutionary trees. Technical report no. 9301, Department of Computer Science, Memorial University of Newfoundland. Available by anonymous ftp from `ftp.cs.mun.ca` in directory `pub/techreports`.

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