Journal of Bioinformatics and Computational Biology
© Imperial College Press

ANCESTRAL MAXIMUM LIKELIHOOD OF EVOLUTIONARY TREES IS HARD

LOUIGI ADDARIO-BARRY

School of Computer Science, McGill University
Montreal, Quebec, Canada
laddar@cs.mcgill.ca

BENNY CHOR

School of Computer Science, Tel-Aviv University
Tel-Aviv, Israel
benny@cs.tau.ac.il

MIKE HALLETT

McGill Centre for Bioinformatics, School of Computer Science, McGill University
Montreal, Quebec, Canada
hallett@mcb.mcgill.ca

JENS LAGERGREN

Stockholm Bioinformatics Center and
Department of Numerical Analysis and Computer Science, KTH Royal Institute of Technology
Stockholm, Sweden
jensl@nada.kth.se

ALESSANDRO PANCONESI

Dipartimento di Informatica, Universitá di Roma "La Sapienza"
Rome, Italy
ale@dsi.uniroma1.it

TODD WAREHAM*

Department of Computer Science, Memorial University of Newfoundland
St. John's, Newfoundland, Canada
harold@cs.mun.ca

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Maximum likelihood (ML) (Neyman, 1971) is an increasingly popular optimality criterion for selecting evolutionary trees. Finding optimal ML trees appears to be a very

*Corresponding author.

hard computational task – in particular, algorithms and heuristics for ML take longer to run than algorithms and heuristics for maximum parsimony (MP). However, while MP has been known to be NP-complete for over 20 years, no such hardness result has been obtained so far for ML.

In this work we make a first step in this direction by proving that ancestral maximum likelihood (AML) is NP-complete. The input to this problem is a set of aligned sequences of equal length and the goal is to find a tree and an assignment of ancestral sequences for all of that tree's internal vertices such that the likelihood of generating both the ancestral and contemporary sequences is maximized. Our NP-hardness proof follows that for MP given in (Day, Johnson and Sankoff, 1986) in that we use the same reduction from VERTEX COVER; however, the proof of correctness for this reduction relative to AML is different and substantially more involved.

Keywords: Maximum Likelihood; Phylogeny Inference; Computational Complexity.

1. Introduction

1.1. *Background*

Most methods for phylogenetic tree reconstruction on n species belong to two categories – the *distance-based* methods (in which the input is a symmetric n-by-n distance matrix) and *character-based* methods (in which the input is an n-by-m matrix of the values of m characters for each of the n species). Given the increasing availability of genomic sequence data, a character matrix typically consists of an m-length multiple alignment of n homologous sequences, one per species. The two most common character-based methods are *maximum parsimony* (MP) and *maximum likelihood* (ML). Each of these methods has many variants, and each has strengths and weaknesses relative to various aspects, *e.g.*, consistency of inference (see Swofford *et al.*[1] and references therein).

An aspect of particular interest is the computational complexity of MP and ML. Each MP and ML variant has a well-defined objective function, and the related decision problems (or at least discretized versions of them) are usually in the complexity class NP. However, the only variants that are known to be solvable in polynomial time are those in which the tree (MP) and the tree and the branch lengths (ML) are given in addition to the data matrix. The goals in these variants are to compute the maximum likelihood of the data given that tree (ML) and the optimal maximum parsimony score and ancestral sequence assignments given that tree (MP) [2,3,4,5,1]. The situation for those variants that must determine optimal trees relative to given data matrices is more problematic – though it has been known for over 20 years that all such MP variants are NP-complete [6,7] (see also Wareham[8] and references), no such results have been found for ML to date. This is particularly frustrating in light of the intuition among practitioners that MP is easier than ML.

In this paper, we make a first step in addressing the complexity of ML by examining the complexity of the ANCESTRAL MAXIMUM LIKELIHOOD (AML) problem [9,10]. This variant is "between" MP and ML in that it is a likelihood method (like ML) but it reconstructs sequences for internal vertices (like MP). In this paper, we show that AML is NP-complete using a reduction from VERTEX COVER that is almost identical to that given for MP by Day, Johnson and Sankoff [7]. Note, however, that the proof of correctness for this reduction relative to AML is different and substantially more involved than that given for MP.

1.2. *Definitions*

In this section we briefly describe the ancestral maximum likelihood (AML) problem. The goal of AML is to find the weighted evolutionary tree, together with assignments to all internal vertices, which is most likely to have produced the observed sequence data. To make this notion meaningful, we must have an underlying substitution model for the process of point mutation. Given such a model, we seek the tree T together with the edge probabilities p_e and sequence assignments s_v for all internal vertices v of T which maximize L, the likelihood of the data.

Evolutionary tree inference methods are usually applied to 4-state (DNA and RNA nucleotide) or 20-state (protein amino acid) data. However, to prove hardness of AML, it suffices to consider the simpler case of the the Neyman 2-state model [11]. In this model, each character of the root is assigned a state according to some initial distribution and each tree-edge has an associated probability $p_e \leq 1/2$ that the character states at the two endpoint vertices of e differ.

For a tree T, let $\mathbf{p} = [p_e]_{e \in E(T)}$ be the edge probabilities and $\psi(1), \psi(2), \psi(3), \ldots, \psi(m) \in \{\mathtt{0}, \mathtt{1}\}^n$ be the observed sequences of length n over m taxa. Given a set $\mathbf{s}$ of sequences of length n labeling the vertices of T, let d_e denote the number of differences between the two sequences labeling the endpoints of the edge $e \in E(T)$. As we are dealing with a symmetric time-reversible model of character change along edges, it is readily seen that the edge probabilities p_e are independent of the position of the root and we can regard T as being unrooted; however, for the purposes of integrating symbol-prior probabilities into the likelihood calculations, we will still designate an arbitrary internal vertex $\mathbf{r}$ as the root.

For a specific edge $e \in E(T)$, the probability of generating the d_e differences and $n - d_e$ non-differences equals $p_e^{d_e}(1-p_e)^{n-d_e}$. As events across different edges are mutually independent, the conditional probability (or the *ancestral likelihood*) of observing ψ given the tree T, the internal sequences $\mathbf{s}$ and the edge probabilities $\mathbf{p}$ is $L(\psi|T, \mathbf{s}, \mathbf{r}, \mathbf{p}) = (\prod_{e \in E(T)} p_e^{d_e}(1-p_e)^{n-d_e}) \times p(\mathbf{r})$, where $p(\mathbf{r})$ is the term produced by multiplying together all of the prior probabilities of each character-state of the sequence assigned to root-vertex $\mathbf{r}$. This conception of AML is called *joint ancestral likelihood* by Pupko *et al.* [12]. We will assume that both states are equally probable in the initial distribution, which makes the root-prior term a constant that can be ignored. This leads to our first definition of ancestral maximum likelihood as an optimization problem:

Ancestral Maximum Likelihood (Version I)
Input: A set S of m binary sequences, each of length n.
Goal: Find a tree T with m leaves, an assignment $p : E(T) \to [0,1]$ of edge probabilities, and a labeling $\lambda : V(T) \to \{0,1\}^n$ of the vertices such that

(1) The m labels of the leaves are exactly the sequences from S, and
(2) $\prod_{e \in E(T)} p_e^{d_e}(1-p_e)^{n-d_e}$ is maximized.

The AML problem may, at first glance, seem like a continuous optimization problem. However, this is not the case given the simplifications made above. Consider an edge probability p_e: Given d_e and the length n of the sequences, the value of p_e that maximizes the individual edge likelihood $p_e^{d_e}(1-p_e)^{n-d_e}$ is simply d_e/n. Upon substituting this value and taking the n-th root, an individual edge likelihood becomes

$$\left(\frac{d_e}{n}\right)^{d_e/n}\left(1-\frac{d_e}{n}\right)^{1-d_e/n} \tag{1}$$

and the logarithm of the tree likelihood expression becomes

$$\sum_{e \in E(T)} \left(\frac{d_e}{n} \log\left(\frac{d_e}{n}\right) + \left(1-\frac{d_e}{n}\right)\log\left(1-\frac{d_e}{n}\right)\right) = \sum_{e \in E(T)} -H_2\left(\frac{d_e}{n}\right), \tag{2}$$

where H_2 is the binary entropy function, $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$. This leads to our second formulation (in the sequel we drop the subscript 2 from logarithms and entropies):

Ancestral Maximum Likelihood (Version II)
Input: A set S of m binary strings, each of length n.
Goal: Find a tree T with m leaves and a labeling $\lambda : V(T) \to \{0,1\}^n$ of the vertices such that

(1) The m labels of the leaves are exactly the sequences from S, and
(2) $\sum_{e \in E(T)} H\left(\frac{d_e}{n}\right)$ is minimized.

The decision version of this formulation is as follows:

Ancestral Maximum Likelihood (Version III)
Input: A set S of m binary strings, each of length n, and a positive number k.
Question: Is there a tree T with m leaves and a labeling $\lambda : V(T) \to \{0,1\}^n$ of the vertices such that

(1) The m labels of the leaves are exactly the sequences from S, and
(2) $\sum_{e \in E(T)} H\left(\frac{d_e}{n}\right) \le k$?

Unless otherwise noted, AML will denote AML-III in the remainder of this paper.

While we show that AML is NP-complete, there are variants of AML that are tractable [9,10,12]. In these variants, we are given the tree and the edge symbol-change probabilities in addition to the input sequences and the goal is to find sequence assignments to inner vertices as to maximize the likelihood. The dynamic programming algorithm given by Pupko *et al.* [12] is particularly elegant and is built along the same lines as dynamic programming algorithms for MP and ML when the tree is given [3,4,5]. Note that our completeness result is derived for the general version of AML given above in which we have to optimize over trees, assignments, and edge probabilities. This much larger parameter space explains the increase in complexity from polynomial time to NP-complete.

2. The Hardness of Ancestral Maximum Likelihood

In this section we establish that AML is NP-complete. We begin by recalling the reduction given in Day, Johnson and Sankoff[7] which establishes the NP-hardness of maximum parsimony. This reduction is defined relative to the following problems:

Vertex Cover (VC)
Input: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.
Question: Is there a subset $V' \subseteq V$ such that $|V'| \leq k$ and for each edge $(u, v) \in E$, at least one of u and v belongs in V'?

Maximum Parsimony (MP)
Input: A set S of m binary strings, each of length n, and an integer $k \geq 0$.
Question: Is there a tree T with m leaves and a labeling $\lambda : V(T) \rightarrow \{0,1\}^n$ of the vertices such that

(1) The m labels of the leaves are exactly the sequences from S, and
(2) $\sum_{e \in E(T)} d_e \leq k$?

Given an instance $\langle G = (V, E), k\rangle$ of VC, the reduction constructs an instance $\langle S, k'\rangle$ of MP such that S is a set of $m = |E| + 1$ strings, each of length $n = |V|$, and $k' = k + |E|$. The first string in S consists of all zeros, *i.e.*,

$$\underbrace{000\ldots00}_{n},$$

and then for every edge $e = (i, j) \in E$ there is a string

$$\underbrace{\overbrace{0\ldots0}^{i-1}1\,\overbrace{0\ldots0}^{j-(i+1)}\,1\,\overbrace{0\ldots0}^{n-(j+1)}}_{n}$$

where only the ith and jth symbols are set to 1. These latter strings are called *edge strings*. Similarly, a string with only the ith symbol set to 1 will be called a *vertex string*. Where the context is clear, we will refer to a vertex labelled with a string in which symbols $\{i_1, i_2, \ldots, i_t\}$ are set to 1 as $i_1 i_2 \ldots i_t$. For example, a vertex labelled with the edge-string in which symbols i and j are set to 1 would be called ij.

The proof in Day, Johnson and Sankoff[7] establishes that optimal solution trees for instances of MP constructed by the reduction above have the following form:

Definition 1. A vertex-labelled tree is *canonical* if it satisfies the following four properties:

(1) For any edge e in T, $d_e = 1$;
(2) The sequence labeling the root of T is the all-zero vector;
(3) The children of the root are internal vertices labelled with vertex strings; and
(4) The children of the internal vertices in (3) are leaves labelled with edge strings.

The following observation is crucial in establishing this fact:

> Given a tree T, suppose there are two leaves ij and kl in T that are connected to a common parent. If the numbers i, j, k, and l are all different then there is no increase in tree-cost if we introduce two new vertices i and k, connect these vertices to the root, and then connect ij to i and kl to k.

The proof concludes by noting that each such canonical tree T defines a vertex cover for the graph G in the given instance of VC – namely, the vertices in G corresponding to the vertex-strings labeling the children of the root in T.

Given the similarity of MP and AML as defined here, it is not surprising that we can re-use the reduction above to show that AML is NP-hard. Indeed, our reduction for AML differs from that given above only in that $k' = (k + |E|)H(\frac{1}{n})$. Unfortunately, we cannot immediately re-use the associated proof of correctness because optimal solution trees for instances of AML constructed by this reduction are not necessarily canonical. This is so because for small values of p, the binary entropy function satisfies

$$H(2p) < 2H(p), \tag{3}$$

which means that in maximal ancestral likelihood trees, it may be cheaper to connect a vertex ij directly to the root, rather than connecting it to a vertex i that is a child of the root. Hence, in our proof, we will use the following relaxed canonical form:

Definition 2. A vertex-labelled tree is *weakly canonical* if it satisfies the following four properties:

(1) For any edge e in T, d_e is either 1 or 2;
(2) The sequence labeling the root of T is the all-zero vector;
(3) The children of the root are internal vertices labelled with either vertex strings or edge strings; and
(4) The children of the internal vertices in (3) are leaves labelled with edge strings.

Given the above, our proof of correctness will be in two parts:

(1) Establish that optimal solution trees for instances of AML constructed by the reduction from instances of VC on arbitrary graphs are weakly canonical.
(2) Define a class of graphs such that (a) VC restricted to this class of graphs is NP-hard and (b) AML optimal solution trees for these restricted VC instances are canonical.

The following definitions will be useful. Given a binary string s, the *weight* of s is the number of 1's in s. Given a vertex v labelled by a sequence s, the *weight* of v is the weight of s. Given a vertex-labelled rooted tree T and an edge $e = (u, v)$ in T, the *cost* of e is the Hamming distance between the sequence-labels of the end-point vertices u and v. Given a vertex-labelled tree T and a vertex v in T, v is a *bad vertex* if the weight of v is greater than 2. Unless otherwise noted, an optimal solution tree for an instance of AML constructed by the reduction will be denoted below as an optimal tree.

Lemma 1. *An optimal tree cannot have bad vertices.*

Proof. Consider an optimal tree T such that T has a minimum number of bad vertices and the sum of the degrees of the bad vertices is minimum. We can assume the following about T:

- *Any leaf in T has weight 2*: Suppose there is a leaf v of weight > 2. One can remove the path in T from v back to either the first vertex with a label from S or the first internal vertex with a descendant-vertex with a label from S; however, this would reduce the number of bad vertices as well as the entropy, violating the choice of T.
- *Any vertex of weight 1 is a child of the root*: One can simply disconnect such a vertex from its parent in T and attach it directly to the root without changing the entropy.
- *Any vertex v of weight > 1 with a weight-2 vertex u as a child must have weight 3*: If this is not the case, the Hamming distance between the sequence-labels of v and u is at least 2; however, this would imply that we could attach u directly to the root without increasing the overall entropy, thereby violating the choice of T.

Let b be a bad vertex in T such that (1) it has maximum distance from the root, (2) it is of minimum weight with respect to (1), and (3) it is of minimum degree with respect to both (1) and (2). By the choice of b, all of its children must have weight 2, which implies that b has weight 3. We will without loss of generality refer to b in the remainder of this proof as ijk.

If ijk has a child attached by an edge of cost 2 or more then that child can be attached directly to the root, violating the choice of T. Therefore ijk can have at most three children, namely ij, jk and ik (see Figure 1). Let us consider the cases of each possible set of children in turn. In what follows let c denote the cost of the edge linking ijk to its parent vertex in T.

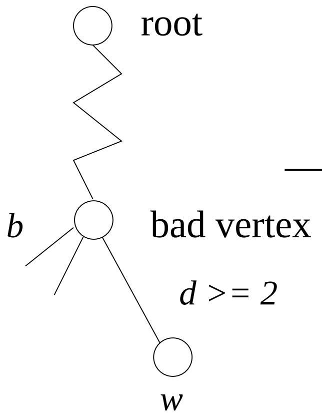

Fig. 1. Proof of Lemma 1: A bad vertex ijk can have at most 3 children.

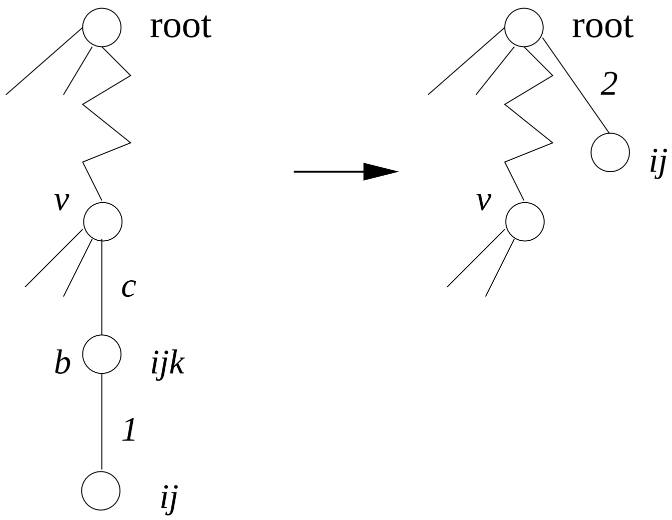

Fig. 2. Proof of Lemma 1, Case 1: Bad vertex ijk has one child.

(1) *ijk has one child*: If ijk has just one child then we can get a tree with lower entropy by removing both ijk and the edges of cost 1 and c and introducing one edge of cost 2 that connects ij directly to the root. This violates the optimality of T, since $H(\frac{c}{n}) + H(\frac{1}{n}) \geq 2H(\frac{1}{n}) > H(\frac{2}{n})$ (see Figure 2).

(2) *ijk has two children*: Suppose ijk has two children ij and jk. One can attach ij and jk to (a possibly new) vertex j, attach j to the root, and delete ijk.

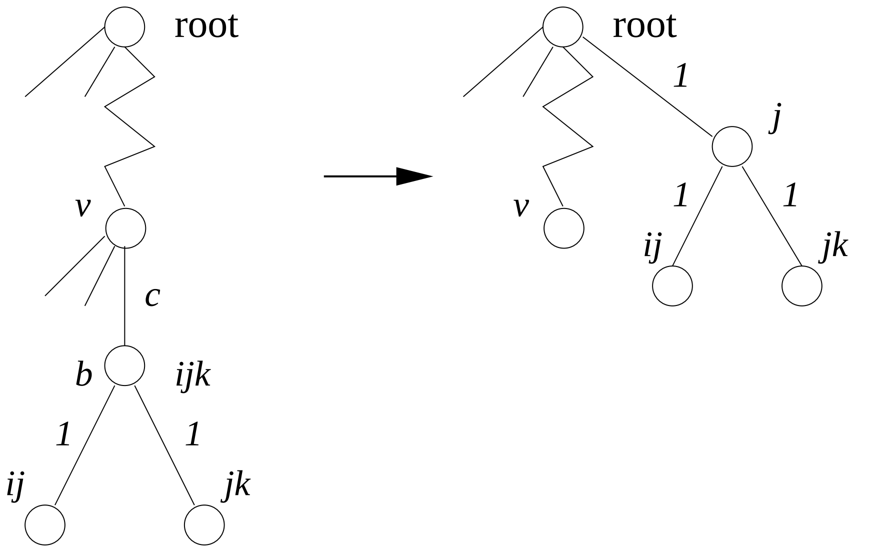

Fig. 3. Proof of Lemma 1, Case 2: Bad vertex ijk has two children.

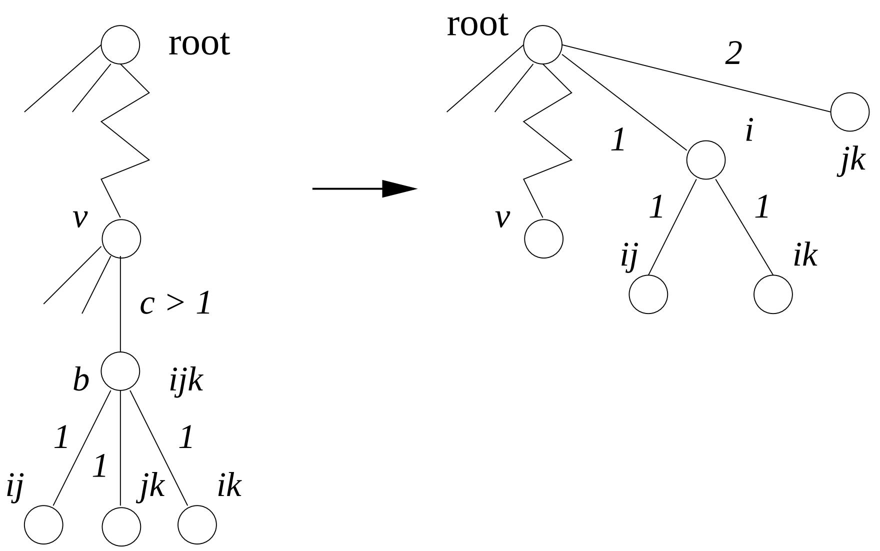

Fig. 4. Proof of Lemma 1, Case 3(a): Bad vertex ijk has three children, cost of edge connecting ijk to parent is > 1.

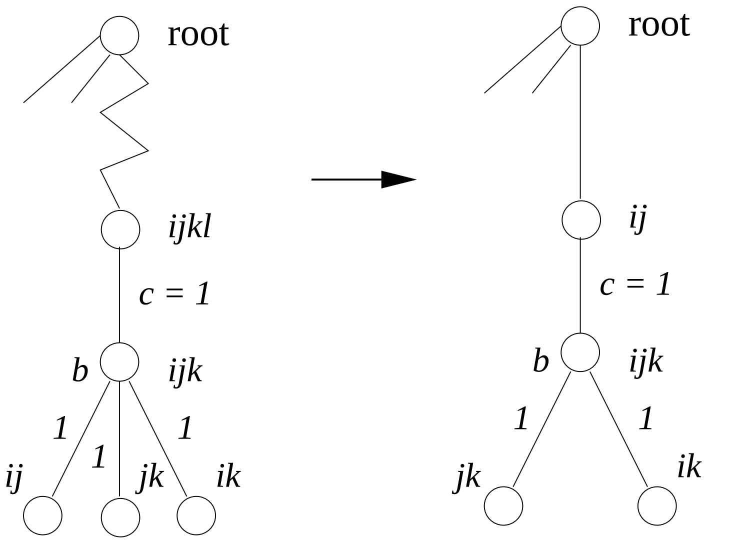

Fig. 5. Proof of Lemma 1, Case 3(b): Bad vertex ijk has three children, cost of edge connecting ijk to parent is 1.

This does not necessarily increase the entropy; however, a bad vertex has been removed, violating the choice of T (see Figure 3).

(3) *ijk has three children*: This has two subcases.

(a) If $c > 1$, ijk can be eliminated as follows: Vertex jk is attached directly to the root, while ij and ik are attached to (a perhaps new) vertex i that is connected to the root. This does not necessarily increase the entropy as $3H(\frac{1}{n}) + H(\frac{2}{n}) \leq 3H(\frac{1}{n}) + H(\frac{c}{n})$; however, a bad vertex has been removed, violating the choice of T (see Figure 4).

(b) If $c = 1, ijk$'s parent must be of weight 4 (as all suitable weight-2 vertices are already children of ijk). Without loss of generality, let this parent be $ijk\ell$. If ijk is its only child we can eliminate $ijk\ell$ by connecting ij directly to the root and making ijk a child of ij. This lowers the entropy and eliminates one bad vertex, thereby violating the optimality of T (see Figure 5). Thus $ijk\ell$ must have a second child, which we denote by b'. By the choice of b, b' both has weight 3 and has 3 children of weight 2. Assume without loss of generality that b' is $jk\ell$. One of the children of $jk\ell$ must be jk, $k\ell$ or $j\ell$; again, without loss of generality, assume this child is jk. This implies that T contains a cycle based on the vertices $ijk\ell$, ijk, $jk\ell$, and jk, which contradicts the acyclicity of T. □

Lemma 2. *An optimal tree cannot have internal vertices of weight* 2.

Proof. We again assume that any vertex of weight 1 is a child of the root. By Lemma 1, we can assume that an optimal tree has no bad vertices. Let T be an optimal tree with the minimum number of weight-2 internal vertices and let ij be one such internal vertex. Any child v of ij must have weight 2 as well; moreover, as the label of v must differ from the label of ij in at least two symbol-positions, the edge connecting ij and c must have cost ≥ 2. Any such child v of ij can thus be attached directly to the root without changing the overall cost of the tree. However, ij would no longer be an internal vertex, violating the choice of T. □

Corollary 1. *An optimal tree is weakly canonical.*

Proof. Lemmas 1 and 2 show that optimal trees consist of vertices of weight ≤ 2 and have no internal vertices of weight 2. Such a tree is weakly canonical by definition. □

The next step is to define a class of graphs for which the associated AML optimal solution trees are canonical. The class of graphs we will use is derived using the graph composition operation.

Definition 3. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the *composition graph* $G_c = G_1[G_2]$ is the graph with vertex set $V = V_1 \times V_2$ and edge set E defined by $E = \{((u_1, u_2), (v_1, v_2)) :$ either $(u_1, v_1) \in E_1$ or $u_1 = v_1$ and $(u_2, v_2) \in E_2\}$.

If G_2 has $h \geq 1$ vertices, the composition graph G_c consists of h isomorphic copies of G_1. Given a vertex u in G_1, let the *copy* u^i *of* u be the vertex corresponding to u in the ith copy of G_1. Let the set of all copies of u in G_c be called the *column* of u and denote this column by U. We will be interested in the class of graphs $G[I_h]$, where G is an arbitrary graph and I_h is the graph on $h \geq 1$ vertices with no edges.

The following general property about minimal vertex covers will be useful in several of the proofs given below.

> *Property 1:* If C is a minimal vertex cover for a graph G, then every $u \in C$ must have a neighbor in G that is not in C.

Indeed, if all neighbors of u were in C then u could be removed from C, violating minimality.

Lemma 3. *For any h,* Vertex Cover *is NP-hard on composition graphs of the form $G[I_h]$.*

Proof. Given a graph G, let $G_c = G[I_h]$. Given a vertex cover C of G, consider a vertex cover C' of G_c consisting of all the copies of vertices of C – that is, if $u \in C$ then $u^i \in C'$, for all i. Such a vertex cover C' is said to be in *normal form.* Note that if C' is in normal form and U is a column then either U is contained in C' or it is disjoint from it.

Let C' be a minimal vertex cover of G_c that is not in normal form. There must therefore be a column U that is neither contained in nor disjoint from C'. Let $u^i \in C' \setminus U$ and let $u^j \in U \setminus C'$. Since C' is minimal, by Property 1, there is a neighbor w of u^i not in C'. However, this would imply that the edge wu^j is not covered, which is a contradiction. Thus, all minimal vertex covers of G_c must be in normal form, including in particular all optimal vertex covers. NP-hardness follows from the fact that we have established a polynomial-time computable bijection between optimal vertex covers of G and of G_c. □

Note that every vertex in any optimal vertex cover of $G[I_h]$ covers at least h edges *uniquely* – that is, it covers at least h edges not covered by any other vertex (this follows from Property 1). This observation will be useful below.

Lemma 4. *An optimal tree associated with an instance of VC based on a composition graph of the form $G[I_h]$ is canonical.*

Proof. Let T be an optimal tree associated with $G_c = G[I_h]$. As T is only known to be weakly canonical by Corollary 1, there is at least one vertex of weight 2 adjacent to the root. Let C be the set of vertices of G_c corresponding to weight-1 vertices adjacent to the root in T. We will now prove that the set of weight-2 vertices directly attached to the root correspond to a matching M in G_c such that $V(M) \cap C = \emptyset$. Suppose that ij and ik are attached to the root. If it is not already there, we introduce i, attach i to the root, and attach ij and ik to i. The new tree has lower entropy since $2H(\frac{2}{n}) > 3H(\frac{1}{n})$. Assume that ij and i are adjacent to the root. We clearly obtain a tree with lower entropy by making ij adjacent to i, instead of the root. It follows that M has the wanted properties. Notice that C is a vertex cover in $G_c \setminus M$.

Consider $uv \in M$. By the above, $u, v \notin C$. Let $v' \neq v$ be a vertex in the column V. Since $uv \in E(G_c)$, v' must be in the vertex cover C. Let us look at the neighbors of v' apart from u. Let w be such a neighbor. Since $v'w$ is an edge, vw is an edge too. This implies that w must then be in C (otherwise, this edge is not covered) and furthermore that all neighbors $w \neq u$ of v' are in C. Hence, the only edge that is covered by v' uniquely is the edge uv' and this holds true for any v' in the column V. However, this implies that there is a much more economical way to cover the edges of $G_c \setminus M$, namely

$$C' = (C \setminus V) \cup \{u\} \ . \tag{4}$$

In terms of entropy of the corresponding trees, this means that if we switch from C to C' as the weight-1 vertices connected to the root of the tree (adding edges in the natural way) then we pay $H(\frac{1}{n})$ but save $H(\frac{2}{n}) + (h-1)H(\frac{1}{n})$, violating the optimality of T. The claim then follows. □

Theorem 1. *AML is NP-complete.*

Proof. AML is clearly in NP. The result then follows from Lemmas 3 and 4. □

3. Concluding Remarks and Future Directions

In this paper, we have shown the NP-completeness of the problem of inferring ancestral sequences under joint maximum likelihood relative to a very simple model of character-change. There are three obvious directions for future research:

(1) *Narrow the gap between tractability and intractability with respect to the status of other variants of AML.* Of particular interest is the version of AML with variable rates across sites. Based on work described in Pupko *et al.*[13], it appears that a jump in complexity occurs when mutation-rate variation across sequence sites is allowed.
(2) *Extend our results for AML to ML under the Neyman 2-state model.*
(3) *Extend our results for AML to ML under more general models.*

Though (2) and (3) remain our ultimate goals, it is not immediately obvious how to extend the present work to attain these goals. A large part of the difficulty is that AML deals with the most likely ancestral-sequence assignment (and hence has that assignment present to be exploited by a reduction) while ML sums likelihoods over all possible ancestral-sequence assignments (leaving only the tree and the likelihood-sum to be exploited by a reduction). It is tempting to believe that the reconstruction of internal vertex sequences is responsible for the computational complexity of MP and AML, and that ML may in fact be easy; however, the NP-hardness of other evolutionary tree inference problems that either do not reconstruct internal states (distance-matrix fitting)[14] or reconstruct internal states in very limited fashions (character compatibility, perfect phylogeny)[15,16] (see also Wareham[8] and references therein) suggest that there are more subtle factors at work here.

In any case, it may be necessary to resort to reductions very different from that given here for AML to show NP-hardness for ML. One promising source of such reductions is various mathematical arguments showing those cases in which ML and other evolutionary tree inference methods give identical results (see Goldman[17] and references therein). Alternatively, one may consider reductions to versions of ML based on different likelihood calculations from those considered here, *e.g.*, calculating likelihood from site pattern probabilities[18,19].

Acknowledgments

We would like to thank the staff of the Bellairs Research Institute of McGill University, where much of the research reported here was done, for their hospitality. The research reported here was supported by ISF grant 418/00 (BC), the EU thematic network APPOL (AP), and NSERC grant 228104 (TW).

References

1. D. Swofford, G. Olsen, P. Waddell and D. Hillis, "Phylogenetic Inference." In D. Hillis, C. Moritz and B. Mable (eds.) *Molecular Systematics* (Second Edition), pages 407–514. Sinauer Associates, Sunderland, MA, 1996.

2. W. Fitch, "Towards defining the course of evolution: minimum change for a specific tree topology," *Systematic Zoology*, **20**, 406–416 (1971).
3. J. Felsenstein, "Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach," *Journal of Molecular Evolution*, **17**, 368–376 (1981).
4. D. Sankoff and R. Cedergren, "Simultaneous comparison of three or more sequences related by a tree." In D. Sankoff and J. Kruskal (eds.) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 253–263. Addison-Wesley Publishing Company, Reading, MA, 1983.
5. D. Swofford and W. Maddison, "Parsimony, Character-State Reconstructions, and Evolutionary Inferences." In R. Mayden (ed.) *Systematics, Historical Ecology, and North American Freshwater Fishes*, pages 186–223. Stanford University Press, 1992.
6. L. Foulds and R. Graham, "The Steiner problem in phylogeny is NP-complete," *Advances in Applied Mathematics*, **3**, 43–49 (1982).
7. W. Day, D. Johnson and D. Sankoff, "The Computational Complexity of Inferring Rooted Phylogenies by Parsimony," *Mathematical Biosciences*, **81**, 33–42 (1986).
8. T. Wareham. *On the Computational Complexity of Inferring Evolutionary Trees.* Technical Report 93-01, Department of Computer Science, Memorial University of Newfoundland, 1993.
9. Z. Yang, S. Kumar and M. Nei, "A new method of inference of ancestral nucleotide and amino acid sequences," *Genetics*, **141**, 1641–1650 (1995).
10. M. Koshi and R. Goldstein, "Probabilistic reconstruction of ancestral protein sequences," *Journal of Molecular Evolution*, **42**, 313–320 (1996).
11. J. Neyman, "Molecular studies of evolution: A source of novel statistical problems." In S. Gupta and Y. Jackel (eds.) *Statistical Decision Theory and Related Topics*, pages 1–27. Academic Press, New York, 1971.
12. T. Pupko, I. Pe'er, R. Shamir and D. Graur, "A Fast Algorithm for Joint Reconstruction of Ancestral Amino Acid Sequences," *Molecular Biology and Evolution*, **17**(6), 890–896 (2000).
13. T. Pupko, I. Pe'er, M. Hasegawa, D. Graur and N. Friedman, "A branch-and-bound algorithm for the inference of ancestral amino-acid sequences when the replacement rate varies among sites: Application to the evolution of five gene families," *Bioinformatics*, **18**(8), 1116–1123 (2002).
14. W. Day, "Computational Complexity of Inferring Phylogenies from Dissimilarity Matrices," *Bulletin of Mathematical Biology*, **49**(4), 461–467 (1987).
15. W. Day and D. Sankoff, "Computational Complexity of Inferring Phylogenies by Compatibility," *Systematic Zoology*, **35**(2), 224–299 (1986).
16. M. Steel, "The complexity of reconstructing trees from qualitative characters and subtrees," *Journal of Classification*, **9**, 71–90 (1992).
17. N. Goldman, "Maximum likelihood inference of phylogenetic trees, with special reference to a Poisson process model of DNA substitution and to parsimony analyses," *Systematic Zoology*, **39**(4), 345–361 (1990).
18. P.J. Waddell and D. Penny, "Evolutionary trees of apes and humans from DNA sequences." In A.J. Locke and C.R. Peters (eds.) *Handbook of Symbolic Computation*, pages 53–73. Clarendon Press, Oxford, 1996.
19. P.J. Waddell, D. Penny and T. Moore, "Extending Hadamard conjugations to model sequence evolution with variable rates across sites," *Molecular Phylogenetics and Evolution*, **8**, 33–50 (1997).

Louigi Addario-Berry received his B.Sc. degree in Computer Science from McGill University, Montreal, PQ, Canada, in 2001. He is currently working on his M.Sc. degree in Computer Science at McGill University.

Benny Chor received his M.Sc. degree in Mathematics from the Hebrew University of Jerusalem, Israel, in 1981 and his Ph.D. degree in Computer Science from MIT, Cambridge, MA, USA, in 1985. He is currently in the School of Computer Science, Tel-Aviv University, Israel.

Mike Hallett received his B.Sc. degree in Computer Science from Queen's University, Kingston, ON, Canada, in 1992 and his Ph.D. degree in Computer Science from the University of Victoria, Victoria, BC, Canada, in 1996. He was a postdoctoral fellow and oberassistant in Computer Science with the Computational Biochemistry Research Group at ETH Zürich in 1996 – 1998 and 1998 – 2000, respectively. In 2000, he joined the School of Computer Science, McGill University, Montreal, PQ, Canada where he is currently the interim director of the McGill Centre for Bioinformatics.

Jens Lagergren is currently in the Department of Numerical Analysis and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. He is also a member of the Stockholm Bioinformatics Center.

Alessandro Panconesi received his M.Sc. and Ph.D. degrees in Computer Science from Cornell University, Ithaca, NY, USA, in 1992 and 1993, respectively. He is currently in the Dipartimento di Informatica (DSI), Universitá di Roma "La Sapienza", Rome, Italy.

Todd Wareham received his M.Sc. degree in Computer Science from Memorial University of Newfoundland, St. John's, NL, Canada, in 1993 and his Ph.D. degree in Computer Science from the University of Victoria, Victoria, BC, Canada, in 1999. He was a postdoctoral fellow with the Computational Biology Group in the Department of Computing and Software, McMaster University, Hamilton, ON, Canada, from 1998 to 1999. He is currently in the Department of Computer Science, Memorial University of Newfoundland.