Maximum Likelihood of Evolutionary Trees is Hard *

Benny Chor † Tamir Tuller ‡

School of Computer Science
Tel-Aviv University

Abstract

Maximum likelihood (ML) is an increasingly popular optimality criterion for selecting evolutionary trees (Felsenstein, 1981). Finding optimal ML trees appears to be a very hard computational task, but for tractable cases, ML is the method of choice. In particular, algorithms and heuristics for ML take longer to run than algorithms and heuristics for the second major character based criterion, maximum parsimony (MP). However, while MP has been known to be NP-complete for over 20 years (Day, Johnson and Sankoff, 1986), reduction from vertex cover), such a hardness result for ML has so far eluded researchers in the field.

An important work by Tuffley and Steel (1997) proves quantitative relations between the parsimony values of given sequences and the corresponding log likelihood values. However, a direct application of it would only give an *exponential time* reduction from MP to ML. Another step in this direction has recently been made by Addario-Berry *et al.* (2004), who proved that *ancestral maximum likelihood* (AML) is NP-complete. AML "lies in between" the two problems, having some properties of MP and some properties of ML. Still, the AML proof is not directly applicable to the ML problem.

We resolve the question, showing that "regular" ML on phylogenetic trees is indeed intractable. Our reduction follows those for MP and AML, but its starting point is an approximation version of vertex cover, known as GAP VC. The crux of our work is not the reduction, but its correctness proof. The proof goes through a series of tree modifications, while controlling the likelihood losses at each step, using the bounds of Tuffley and Steel. The proof can be viewed as correlating the value of any ML solution to an arbitrarily close approximation to vertex cover.

Key words: Maximum likelihood, tree reconstruction, maximum parsimony, intractability, approximate vertex cover.

1 Background

Molecular data, and even complete genomes, are being sequenced at an increasing pace. This newly accumulated information should make it possible to resolve long standing questions in evolution, such as reconstructing the phylogenetic tree of placental mammals and estimating the times of species divergence. The analysis of this data flood requires sophisticated mathematical tools and algorithmic techniques. Two character-based methods are widely used in practice: MP (*maximum parsimony*, Fitch, 1971 [11]) and ML (*maximum likelihood*, Felsenstein, 1981 [8]). It is known that ML is *consistent*, namely with high probability, for long enough input sequences, the correct tree is the tree maximizing the likelihood [10, ch. 16]. Consistency does not hold for MP,

*Research supported by ISF grant 418/00

†benny@cs.tau.ac.il

‡corresponding author, tamirtul@post.tau.ac.il

and in fact for certain families of trees (the so called *Felsenstein zone* [9]) MP will reconstruct the *wrong* trees, even for arbitrarily long input sequences. The two methods are known to be computationally intensive, and exact algorithms are limited to just about $n = 20$ sequences. This forces practitioners to resort to heuristics. For both exact algorithms and heuristics, ML seems a harder problem than MP.

In the absence of concrete lower bound techniques, the major tool for demonstrating computational intractability remains NP hardness proofs. Both MP and ML have well-defined objective functions, and the related decision problems (or at least discretized versions of them) are in the complexity class NP. It has been known for over 20 years that MP is NP-complete [5, 12, 4, 6, 18], (see also [22] and references), using an elegant reduction from vertex cover (VC). However, no such result has been found for ML to date. This is particularly frustrating in light of the intuition among practitioners that ML is harder than MP.

Tuffley and Steel have investigated the quantitative relations between MP and ML [21]. In particular, they showed that if the n sequences are padded with sufficiently many zeroes, the ML and MP trees coincide. Since parsimony is invariant under padding by zeroes, this approach could in principle lead to a reduction from MP to ML. Unfortunately, the upper bound provided in [21] on the padding length is *exponential* in n. A step in a different direction was taken by Addario-Berry *et al.* [1]. They studied the complexity of AML (Ancestral Maximum Likelihood) [16, 23]. This variant of ML is "between" MP and ML in that it is a likelihood method (like ML) but it reconstructs sequences for internal vertices (like MP). They showed that AML is NP-complete, using a reduction from (exact) Vertex Cover.

Our NP hardness proof of ML uses ingredients from both [21] and [1], as well as new insights on the behavior of the likelihood function on trees. The reduction itself is essentially identical to that given for MP by Day, Johnson, and Sankoff [5], and also used in the AML paper. However, our starting point is not *exact* VC but the *gap* version of it [2, 15]. The proof of correctness for this reduction relative to ML is different, and substantially more involved. We define a family of *canonical trees*. Every such tree is associated with a unique cover in the original graph. We show that if L is the likelihood of the canonical tree, n is the number of vertices in the original graph, m is the number of edges in the original graph, and c is the size of the associated cover, then as $n \to \infty$,

$$\frac{-\log(L)}{(m+c)\log(n)} \to 1 \ .$$

In particular, this gives an inverse relation between likelihood and cover size: Larger L implies smaller c, and vice versa.

When proving the correctness of the reduction, we want to establish two directions: ($\Rightarrow$) If the original graph has a small cover, then there is a tree with high likelihood, and ($\Leftarrow$) that the existence of a tree with high likelihood implies the existence of a small cover. The first direction is easy, using the canonical tree related to the small vertex cover. It is the other direction that is hard, because there is no obvious relation between the log likelihood of a *non-canonical* tree and the size of any cover. What we do, starting from any ML tree, is to apply a sequence of modifications that leads it to a *canonical tree*. The whole series of modifications may actually *decrease* the likelihood of the resulting, canonical tree vs. the original, ML one. We use the techniques of [21] to infer likelihood properties from parsimony ones, and show that in every step, the log likelihood decreases by at most $O(\log n)$ bits. Here we rely on the fact that at each step we only modify a small size sub-forest ($2 \log \log n$ leaves at most). Finally, we show that the total number of modifications is not too large – at most $n/\log \log n$. This allows us to show that the ratio of the log likelihood of the last, canonical tree, and the log likelihood of the ML tree, approaches 1 as $n \to \infty$. This proves that log ML is tightly related to an approximate vertex cover, establishing NP hardness of ML.

The size of the subforests had to be carefully balanced between two conflicting demands. On one hand, we want to make sure that at every step, the log likelihood loss is small. As will be explained in detail later, this loss is $O(s^2 + \log n)$, where s is an upper bound on the subforests' sizes. We want to keep the loss no larger than $O(\log n)$, implying s must be $O(\log \log n)$. On the other hand, it is important to keep the overall number of steps $o(n)$, to keep the accumulated loss $o(n \log n)$. This implies that subforests sizes cannot be too small, like for example $O(1)$.

2 Proof's Overview

In this section we give a high level description of the hardness proof. The reduction is from the GAP VERTEX COVER problem on graphs whose degree is at most three, a problem proved NP-hard in 1999 by Berman and Karpinski [2, 15].

Given a graph $G = (V, E)$ of max degree 3 with $n = |V|$ nodes and $m = |E| \leq 1.5n$ edges, we construct an ML instance, consisting of $m + 1$ binary strings of length n. The goal is to find a tree with the $m + 1$ sequences at its leaves, and an assignment of substitution probabilities to the edges of that tree (edges' length), such that the likelihood of generating the given sequences is maximized. The proof relates the approximate max log likelihood value to the size of a vertex cover in G. This approximation is tight enough to enable solving the original gap problem.

Our reduction follows the one for maximum parsimony given by Day, Johnson and Sankoff [5] and for ancestral ML, given by Addario-Berry *et al.* [1]. Both reductions were from the (exact) VERTEX COVER problem. In this reduction we generate one string with only 0s, and m "edge strings" that contain exactly two 1s each, and naturally encodes an edge. Consider all unrooted weighted trees with $m + 1$ leaves that have the given sequences at their leaves. We say that such tree is in *canonical form* if the following properties hold (see figure 1):

Definition 2.1

1. *There is an internal node (called the* "root" *for clarity, even though the trees are unrooted) that has the all zero leaf as a son, and the length of the edge going to this leaf is* 0.
2. *All leaves are one or two tree edges away from the root.*
3. *If a leaf is two tree edges away from the root, then the subtree that contains that leaf has two or three leaves. In this case, all two or three sequences at the leaves share a "*1*" in the same position.*

Canonical trees "uniquely" define a vertex cover, where each subtree corresponds to one, two, or three original edges that are covered by one node. (For the subtrees with one leaf, the covering vertex can correspond to either end point, while for size two and three subtrees, the covering vertex is uniquely defined.) Consequently, given a tree in canonical form, we can quantify the size of the corresponding vertex cover of the original graph. The reason we force the root to be connected to the all zero leaf with an edge of weight 0 is that this way the root itself is "effectively forced" to the all zero label (with probability 1). This enables us to express the likelihood of a canonical form tree as a product of the likelihoods of its subtrees. In particular, there is no influence, or dependency, between different subtrees.

The major part of the proof is showing that given any ML tree, T_{ML}, with the given "reduction sequences" at its leaves, there is a series of local modifications on trees with the given sequences at their leaves, such that in each modification the log likelihood of the resulting tree is decreased by at most $O(\log n)$ per step, and the final tree, T_{Ca}, is in canonical form. The number of modifications is $o(n)$, which is small enough to establish a tight ratio $1 - o(1)$ between the max

log likelihood and the log likelihood of the final, canonical tree. In each step, we transform one tree to another. We identify a small subforest, containing between $\log\log n$ and $2\log\log n$ leaves. Such a subforest is a union of subtrees with a common internal node (not the "root"). Using the bound on the degree of the original graph, we show that the parsimony score of this subforest when its root is labeled by the all zero string can be worse by at most a constant $B < 8^4$ than the score with any other root labeling. Using the results of Tuffley and Steel [21], and the small size of the subtree, it is possible to unroot this subforest, rearrange it, and connect it directly to the root in a "canonical way", such that the overall log likelihood of the whole tree decreases by at most $B\log n + o(\log n)$. Over the series of $n/\log\log n$ modifications, the overall decrease is at most $Bn\log n/\log\log n + o(n\log n/\log\log n)$. We show that the log likelihood of the final canonical tree, T_{Ca}, is $\theta(n\log n)$. This is sufficiently large to show that despite such decrease,

$$\frac{\log L(S|T_{Ca})}{\log L(S|T_{ML})} = 1 - o(1) \ .$$

Every tree in canonical form naturally corresponds to a vertex cover in the original graph. The tight relation between $\log L(S|T_{ML})$ and $\log L(S|T_{Ca})$ implies a tight relationship between the size of an approximate vertex cover in the original graph and the maximum likelihood tree on the given sequences, and establishes the NP hardness of maximum likelihood on phylogenetic trees.

3 Model, Definitions and Notations

In this section we describe the model and basic definitions that we will use later. These definitions include phylogenetic trees and characters, the parsimony score, Neyman's two state model, and the likelihood function. In most of this paper, we assume that characters have one of two states, 0 or 1. Let $S = [s(1), s(2), s(3), \ldots, s(n)] \in \{0,1\}^{n\times k}$ be the observed sequences of length k over n taxa (on n leaves). Given such sequences, both the maximum parsimony and the maximum likelihood criteria aim at finding the tree (or trees) that "best explain" this data. Each uses a different objective function. In this section, both are defined and explained.

Definition 3.1 *(Phylogenetic trees, characters [21])*
A phylogenetic tree with n leaves is a tree $T = (V(T), E(T))$ with no degree two vertices, such that each leaf (degree one vertex) is given a unique label from $[n] = \{1, ..., n\}$. For convenience, we identify each leaf with its label. A non leaf vertex is called an internal vertex. A function $\lambda : [n] \to \{0,1\}$ is called a state function for T. A function $\hat{\lambda} : V(T) \to \{0,1\}$ is called an extension of λ on T if it coincides with λ on the leaves of T. In a similar way we define a functions $\lambda^k : [n] \longmapsto \{0,1\}^k$ and an extension $\hat{\lambda}^k : V(T) \longmapsto \{0,1\}^k$. This later function is called a labelling of T. If $\hat{\lambda}^k(v) = s$ we say that the string s is the labelling of the vertex v.

Given a labelling $\hat{\lambda}^k$, let $d_e(\hat{\lambda}^k)$ denote the number of differences between two the labellings of the endpoints of the edge $e \in E(T)$.

Definition 3.2 *(Maximum parsimony score)*
Let T be a tree with n leaves, and S be a set of n binary strings, all of length k strings. Let ${\lambda^k}_{pars} : [n] \longmapsto S$ be an onto labeling of T's leaves by S' strings. Let $\hat{\lambda}^k_{pars} : V(T) \longmapsto \{0,1\}^k$ be an extension of λ^k that minimizes the expression $\sum_{e\in E(T)} d_e(\hat{\lambda}^k)$. We define the parsimony score for S, T, λ^k, $pars(S, T, \lambda^k)$, as the value of this sum. A maximum parsimony tree (or trees) for the set of binary strings, S, is a tree (or trees) and leaf labeling thats minimizes the sum above over all trees T and assignments λ^k of the strings in S to their leaves. The value of the sum on this tree is called the parsimony score for the set of strings S.

When the labeling $\hat{\lambda}^k$ is clear, we simply use d_e instead of $d_e(\hat{\lambda}^k)$. In the likelihood setting, we endow edges with "mutation" probabilities. For a tree T, let $\mathbf{p} = [p_e]_{e \in E(T)}$ be the edge probabilities. We use the Neyman two states model [17]. Given labels of length k, each position $j \in \{1, \ldots, k\}$ is called a *site*. According to this model:

- Leaves' labels are strings from $\{0,1\}^k$.
- The "edge probability" p_e satisfies $0 \leq p_e \leq \frac{1}{2}$.
- The probability of a net change of state (from '1' to '0' or vice versa) occurring across an edge e (a "mutation event") is given by p_e (the "length" of edge e).
- Mutation events on different edges are independent.
- Different sites mutate independently.

The likelihood of observing an $S \in \{\mathtt{0},\mathtt{1}\}^{n \times k}$, given the tree T with $r \leq n-2$ internal nodes and the edge probabilities $\mathbf{p}$, $L(S|T,\mathbf{p})$, is defined as

$$L(S|T,\mathbf{p}) = \prod_{i=1}^{k} \sum_{\mathbf{a} \in \{\mathtt{0},\mathtt{1}\}^r} \prod_{e \in E(T)} m(p_e, S_i, a_i) \ , \tag{1}$$

where $\mathbf{a}$ ranges over all combinations of assigning characters states ($\mathtt{0}$ or $\mathtt{1}$) to the r internal nodes of T. This notion of ML is termed maximum *average* likelihood in Steel and Penny [20]. Each term $m(p_e, S_i, a_i)$ is either p_e or $(1-p_e)$, depending on whether in the i-th site of S and $\mathbf{a}$, the two endpoints of e are assigned different characters states (and then $m(p_e, S_i, a_i) = p_e$) or the same characters states (and then $m(p_e, S_i, a_i) = 1 - p_e$). The ML solution(or solutions) for a specific tree T is the point (or points) in the edge space $\mathbf{p} = [p_e]_{e \in E(T)}$ (where $0 \leq p_e \leq 1/2$) that maximizes the expression $L(S|T,\mathbf{p})$. The global ML solution(or solutions) is the pair (or pairs) $(T,\mathbf{p})$, maximizing the likelihood over all trees T of n leaves and all edge probabilities $\mathbf{p}$, see [8], Steel [19], and Tuffley and Steel [21] for more details. It is easy to see that by site independence, an equivalent way to define the likelihood of observing S in the tree T is:

$$L(S|T,\mathbf{p}) = \sum_{\lambda \in \{0,1\}^{k \times r}} \prod_{e \in E(T)} p_e^{d_e(\lambda)} \cdot (1 - p_e(\lambda))^{k - d_e(\lambda)} \tag{2}$$

In the rest of the paper we use this definition for likelihood.

4 Properties of Maximum Likelihood Trees

In this section we prove some useful properties of ML trees. We start with properties of general trees and continue with canonical ones.

4.0.1 General Properties of ML Trees

In our NP-hardness proof we want to show that the ML tree for a set of strings, which will be described on the next section, have log likelihood arbitrarily close to a tree of the canonical form. We achieve this by continuously pruning sub-forests that satisfy certain conditions, and rearranging them in a canonical way around a certain internal node. Theorem 4.4 describes bound on the decrease in the log likelihood by such rearrangements.

The following Lemma is used several times in the rest of this paper.

Lemma 4.1 *Let T be a phylogenetic tree with edge probabilities* $\mathbf{p}$*, let S (set of binary string of length k) denote the labelling for the leaves of the tree. Suppose F_1 and F_2 are two disjoint subforests that have x as their common root, and their union is all of T. Let S_1 and S_2 be the leaf labelling of F_1 and F_2, respectively. Let ℓ_x denote labelling of x. Then the likelihood of observing S given T and p is:*

$$L(S|T,p) = \sum_{s\in\{0,1\}^k} L(S_1, l_x = s|F_1, p) \cdot L(S_2, l_x = s|F_2, p).$$

Proof. Follows directly from equation (2). ■

For "standard" phylogenetic trees, the internal nodes do not have any specified labelling, or state, while leaves are labelled by a k long sequence. In the course of our modifications we could have a leaf with no labelling (see figure 2).

The next Lemma states that such "unlabelled" leaves can be pruned without effecting the likelihood.

Lemma 4.2 *Let T be a phylogenetic tree with an unlabelled leaf. By pruning this leaf (and the edge connecting it to its ancestor) we will get a tree, T', with equal likelihood.*

Proof. Let $S = \{S_i\}$ be the set of leaves' labels (binary strings of length k), let h be the unlabelled leaf, let h' be its ancestor, according to our model:

$Pr(S|T,p) = \sum_{x\in\{0,1\}^k} Pr(S, l_h = x|T,p) = \sum_{x\in\{0,1\}^k} \sum_{y\in\{0,1\}^k} Pr(S, l_{h'} = y|T',p) \cdot Pr(l_h = x, l_{h'} = y|p)$

and since $\sum_{x\in\{0,1\}^k} Pr(l_h = x, l_{h'} = y|p) = 1$ we get:

$\sum_{x\in\{0,1\}^k} \sum_{y\in\{0,1\}^k} Pr(S, l_{h'} = y|T',p) \cdot Pr(l_h = x, l_{h'} = y|p) = \sum_{y\in\{0,1\}^k} Pr(S, l_{h'} = y|T',p) = Pr(S|T',p)$

■

Lemma 4.3 *Let T be a phylogenetic tree with an internal node, h, of degree two, and let x, y be its neighbors. Then h can be eliminated to create an (x,y) edge without changing the likelihood.*

Proof. Let $p_{h,x}$ and $p_{h,y}$ be the mutation probabilities across the edges (h,x) and (h,y) respectively. Choose $p(x,y) = p_{h,x} \cdot (1-p_{h,y}) + p_{h,y} \cdot (1-p_{h,x})$. It is easy to see that for $0 \le p_{h,x}, p_{h,y} \le 1$, we get $0 \le p_(x,y) \le 1$, and that the mutation probability across the path between x to y does not change. ■

Theorem 4.4 *Let T^+ be the tree consisting of T_{new}, hung off a node at distance 0 from the all zero leaf. Let T^- be a tree where $T_1, .., T_j$ hang off a common root h that is labelled by the length k sequence z Suppose there is W such that for every labelling s of h, W times the likelihood of T^- is less than the likelihood of T^+. Let $T_{original}$ be the original tree, and $T_{arranged}$ the tree resulting from uprooting $T_1, .., T_j$ arranging them to T_{new} (the subtrees $T_1, .., T_j$ and the tree T_{new} are over the same taxa) and hanging them off a node at distance zero from an all zero leaf. Then the likelihood of $T_{arranged}$ is at least as large as W times the likelihood of $T_{original}$.*

Proof. The likelihood of labelling S, given $T_{original}$, the initial tree, is (see Lemma 4.1):

$$L_1 \equiv L(S|T_{original}) = \sum_z L(S_1, l_h = z|T^-) \cdot L(S_2, l_h = z|T_{original} \setminus T^-).$$

The likelihood of S given $T_{arranged}$ is

$$L_2 \equiv L(S|T_{arranged}) = \sum_z L(S_1, l_h = 0^k|T_{new}) \cdot L(S_2, l_h = z|T_{arranged} \setminus T^-).$$

And according to our assumption:

$$\forall_z (L(S_1, l_h = 0^k|T_{new}) \geq W \cdot L(S_1, l_h = z|T^-)) \ .$$

■

The following corollary is direct result from theorem 4.4

Corollary 1 *Let $\ell f(T^-)$ denote the strings at the leaves of T^- and z denote labelling of ℓ_h, let $T^*(z)$ denote the tree structure of $\ell f(T^-) \cup z$ with largest likelihood, let $z*$ denote z such that $\forall_z L(\ell f(T^-) \cup z|T^*(z)) \leq L(\ell f(T^-) \cup z*|T^*(z*))$. If*

$$L(\ell f(T^-), l_h = 0^k|T_{new}) \geq W \cdot L(\ell f(T^-) \cup z*|T^*(z*)),$$

then

$$L(S|T_{arranged}) \geq W \cdot L(S|T_{original}).$$

Lemma 4.5 *Let T^S_{ML} denote the ML tree for the set of taxa S. For any $S' \subseteq S$ there is $T^{S'}_{ML}$ such that $L(S'|T^{S'}_{ML}) \geq L(S|T^S_{ML})$.*

Proof. Suppose $S = S' \cap x$, let y denote the neighbor internal vertex of x in T^S_{ML}, let $0 \leq p_{x,y} \leq 1$ denote the edge probability of (x, y), let $T^{S'}_{ML}$ denote the resultant tree after removing x and $p_{x,y}$ from T^S_{ML}. The proof follows from the following relations:

$$L(S|T^S_{ML}) = \sum_{\ell_y = 0,1^n} L(S' \cup y = \ell_y|T^{S'}_{ML}) \cdot p_{x,y}^{|\{i:y_i \neq x_i\}|} \cdot (1 - p_{x,y})^{|\{i:y_i = x_i\}|}$$

$$L(S'|T^{S'}_{ML}) = \sum_{\ell_y = 0,1^n} L(S' \cup y = \ell_y|T^{S'}_{ML})$$

■

Lemma 4.6 *Suppose there are positions where the value of all the strings in $\ell f(T^-)$ (see corollary 1) is zero, then the value of $z*$ (see corollary 1) is also zero in these positions.*

Proof. Let $T^*(\overline{z*})$ denote the resultant tree after removing $z*$ and the edge which include $z*$ from $T^*(z*)$. By lemma 4.5 the likelihood of $L(\ell f(T^-) \cup z*|T^*(z*)) \leq L(\ell f(T^-)|T^*(\overline{z*}))$. It is easy to see that by choosing $z*$ which is equal to one of the stings in $\ell f(T^-)$, let S_a denote the string, and connecting it as a leave at distance zero from S_a (namely if S_b is the neighbor of S_a and $z*$ thus $p_{S_b,S_a} = p_{S_b,z*} = 0$), we achieve this upper bound. ■

For any tree T on n leaves and any observed sequences S, we denote by $\mathbf{p}^* = \mathbf{p}^*(S, T)$ the edge probability that maximize $L(S|T, p)$. The following Theorem is a restatement of Theorem 7 from the work of Tuffley and Steel [21].

Theorem 4.7 *Let S be a set of binary strings of length $k = k_c + k_{nc}$, where k_c is the number of constant characters in S (i.e. positions that have the same value for all the strings). Then for large enough k_c, the maximum likelihood and the maximum parsimony tree for S are identical. Furthermore, for every tree T:*

$$2^{-\log(k_c)\cdot pars(S,T)-C^d_{T,,pars(S,T)}} \leq L(S|T) \equiv Pr(S|p^*,T) \leq 2^{-\log(k_c)\cdot pars(S,T)-C^u_{T,,pars(S,T)}}$$

and

$$\lim_{k_c\to\infty} \frac{-\log(Pr(S|p^*,T))}{\log(k_c)} = pars(S,T)$$

where $C^u_{T,pars(S,T)}$ and $C^d_{T,pars(S,T)}$ are subquadratic functions of the size of $|V(T)|$ (the number of vertices in the tree) and of $pars(S,T)$ (the parsimony score for the tree given S). They do not depend on the number of constant sites, k_c (notice that by increasing k_c we do not change $pars(S,T)$).

Corollary 2 *Let T_a and T_b be two tree topologies for a string set S. S Contain n sequences of length k. Suppose that the strings in S have a large enough number of constant characters, k_c,* i. e. *k_c is doubly exponential in n, $pars(S,T_a)$, and $pars(S,T_b)$. If $pars(S,T_a) < pars(S,T_b)$, then $Pr(S|p^*,T_a) > Pr(S|p^*,T_b)$ (Notice that changing k_c does not effect the parsimony score).*

We remark that in general equality in the parsimony score does not imply equality in the likelihood. The next corollary generalizes the previous one to general trees with one an internal node that is labelled.

Corollary 3 *Let S be a string set with length k strings. Suppose that S has k_c constant positions. The likelihood, $Pr(S_1, l_h = z|p^*, F)$, of a subforest F with r subtrees $T_1,..,T_r$ and with a label $l_h = z$ at the root of the subforest (see figure 3) is*

$$Pr(S, l_h = z|p^*, F) = \prod_{i=1}^{r} Pr(S^i_1, l_h = z|p^*, T^i)$$

thus:

$$Pr(S, l_h = z|p^*, F) \geq \prod_{i=1}^{r} 2^{-log(k_c)\cdot pars(S^i\cup z,T_i)-C^d_{T_i,pars(S^i\cup z,T_i)}},$$

and

$$Pr(S, l_h = z|p^*, F) \leq \prod_{i=1}^{r} 2^{-log(k_c)\cdot pars(S^i\cup z,T_i)-C^u_{T_i,pars(S^i\cup z,T_i)}},$$

where $C^u_{T_i,pars(S^i\cup z,T_i)}$ and $C^d_{T_i,pars(S^i\cup z,T_i)}$ are functions for the sub tree T_i with the strings set $S^i \cup z$, as was defined in Theorem 4.7. Let $pars(S \cup z, F) = \sum_i pars(S^i \cup z, T_i)$, and let $C^u_{S\cup z,F} = \sum_i C^u_{T^i,pars(S^i\cup z,T_i)}$ and let $C^d_{S\cup z,F} = \sum_i C^d_{T^i,pars(S^i\cup z,T_i)}$. Then

$$2^{-log(k_c)\cdot pars(S\cup z,F)-C^d_{S\cup z,F}} \leq Pr(S, l_h = z|p^*, F) \leq 2^{-log(k_c)\cdot pars(S\cup z,F)-C^u_{S\cup z,F}}.$$

Proof. The proof follows directly from Theorem 4.7 and the properties of our model. ■

Corollary 4 *Let T_1 and T_2 be two subforests with the same number of leaves, each with its common ancestor as described in corollary 3. Let $Pr(S_1, l_{h1} = z_1|p^*_1, T_1)$ and $Pr(S_2, l_{h2} = z_2|p^*_2, T_2)$ be the maximum likelihood scores of S_1, S_2, for T_1 and T_2, respectively. We fix two labels at the*

roots of T_1,T_2 (not necessarily the same label). If under this setting the parsimony scores of the two subforests are equal, then the likelihood ratio is "sandwiched" between two functions C_1, C_2:

$$C_1 \leq \lim_{k_c \longrightarrow \infty} \frac{Pr(S_2, l_{h2} = z_2 | p_2^*, T_2)}{Pr(S_1, l_{h1} = z_1 | p_1^*, T_1)} \leq C_2$$

where C_1, C_2 are sub quadratic functions of $|V(T_1)|$, $|V(T_2)|$, $pars(S_1, T_1)$, and $pars(S_2, T_2)$.

Proof. The proof follows from Theorem 4.7 and corollary 3. We omit the details. ■

4.0.2 Properties Related to Canonical ML Trees

We now study properties related to canonical ML trees (definition 2.1), which play an important role in our reduction.

Definition 4.8 *Let T_{C_i} ($i = 1,2,$ or 3) be a phylogenetic tree with $i+1$ leaves, and one internal node (*i.e. *T_{C_i} has the star topology). Suppose one of the strings in the leaves is the all zero string (of length $k = n$). The other i strings are all of weight 2 (two 1s), and for $i > 1$ they all share one "1" position. Let $ML_i(n)$ be the log ML score of T_{C_i}. Let S_{C_i} denote the strings in the leaves of tree T_{C_i} (see figure 4).*

It is easy to see that $ML_i(n)$ does not depend on the specific choice of strings in T_{C_i}

Lemma 4.9 *Let C_1^u, C_2^u, C_3^u, C_1^d, C_2^d, C_3^d denote constants, then for n large enough the following properties hold:*

1. $-2 \cdot \log(n) + C_1^d \leq ML_1(n) \leq -2 \cdot \log(n) + C_1^u$
2. $-3 \cdot \log(n) + C_2^d \leq ML_2(n) \leq -3 \cdot \log(n) + C_2^u$
3. $-4 \cdot \log(n) + C_3^d \leq ML_3(n) \leq -4 \cdot \log(n) + C_3^u$

Proof. The proof follows from Theorem 4.7, corollary 4, and direct calculations. ■

Theorem 4.10 *Let T_a and T_b be two canonical trees with $m+1$ leaves labelled by S, where S contains strings of length k. Let d_a and d_b denote the degree of the root of these trees, respectively. Let p_a^* and p_b^* be optimal edges weights for these trees, respectively. Then*

$$\log(P(S|p_b^*, T_b)) = -(d_b + m) \cdot \log n + d_b \cdot C_b$$

and

$$\log(P(S|p_a^*, T_a)) = -(d_a + m) \cdot \log n + d_a \cdot C_a \ ,$$

where $C_b = \theta(n)$, and $C_a = \theta(n)$.

So for large enough n:

$$lim_{n \to \infty} \frac{\log(P(S|p_a^*, T_a))}{\log(P(S|p_b^*, T_b))} = \frac{d_a + m}{d_b + m}.$$

In particular if $d_a = d_b$ then:

$$lim_{n \to \infty} \frac{\log(P(S|p_a^*, T_a))}{\log(P(S|p_b^*, T_b))} = 1.$$

Proof. According to Lemma 4.9 the log likelihood of a subtree, T_{c_i} (that is hung off the root) is $ML_i(n) = (i+1)\cdot\log(n) + C_i = i\cdot\log(n) + \log(n) + C_i$, where i is the number of leaves (that are not $\neq 0$) in the subtree, $C_i^d \leq C_i \leq C_i^u$, where C_i^d, C_i^u are the constant from Lemma 4.9. Let T_x be a canonical tree with degree d_x at its root. Since all the subtrees together have m leaves other than the all zero leaf, the log likelihood of this tree is (summing the log likelihood of all its d_x subtrees) $d_x\cdot\log(n) + m\cdot\log(n) + C_x$. Where

$$d_x\cdot\min\{C_1^d, C_2^d, C_3^d\} \leq C_x = \sum_{T_i\in T_x} C_i \leq d_x\cdot\max\{C_1^u, C_2^u, C_3^u\}.$$

Thus the log likelihood of T_a and T_b is $(m+d_a)\cdot\log(n) + d_a\cdot C_a$, and $(m+d_b)\cdot\log(n) + d_b\cdot C_b$, respectively, where
$d_a\cdot\min\{C_1^d, C_2^d, C_3^d\} \leq C_a \leq d_a\cdot\max\{C_1^u, C_2^u, C_3^u\}$, $d_b\cdot\min\{C_1^d, C_2^d, C_3^d\} \leq C_a \leq d_b\cdot\max\{C_1^u, C_2^u, C_3^u\}$.
Thus the log likelihood ratio of these trees is

$$\frac{\log(P(S|p_a^*, T_a))}{\log(P(S|p_b^*, T_b))} = \frac{-(d_a+m)\cdot\log n + d_a\cdot C_a}{-(d_b+m)\cdot\log n + d_b\cdot C_b}$$

since $n \leq m \leq 1.5\cdot n$ and $\frac{m}{3} \leq d_b, d_a \leq m$. ■

5 NP-Hardness of Maximum Likelihood

The decision version of maximum likelihood is the following:

Problem 5.1 *Maximum likelihood, (ML)*
Input: *S, A set of binary strings, all of length k, and a negative number L.*
Question: *Is there a tree, T, such that* $\log(Pr(S|p^*(S,T),T)) > L$ *?*

A gap vertex cover problem is the following:

Definition 5.2 *Gap problem for vertex cover, $gap-VC[C_1, C_2]$*
Input: *A graph, $G=(V,E)$, two positive numbers, C_1 and C_2.*
Task: *Does G have a vertex cover smaller than C_1? Or is the size of each vertex cover is larger than C_2 ? (If the minimum vertex cover is in the intermediate range, there is no requirement.)*

Our proof uses a reduction from the gap version of vertex cover, restricted to degree 3 graphs, to maximum likelihood. We use the following hardness result of Karpinski and Berman [2].

Theorem 5.3 *[2] The following problem, $gap-VC_3[\frac{144}{284}\cdot n, \frac{145}{284}\cdot n]$, is NP-hard: Given a degree 3 graph, G, on n nodes, is the minimum VC of G smaller than $\frac{144}{284}\cdot n$? Or is it larger than $\frac{145}{284}\cdot n$?* [1]

We reduce the version of $gap-VC_3$ above to ML.

[1] We could also use the deep gap VC results of Håstad [13]) and Dinur and Sufra [7]. However their graphs are of bounded degree greater than 3 and it seems that the modification to bounded degree 3 graphs would yield smaller gaps (not effecting the hardness of ML, though).

5.1 Reduction and Proof Outline

Given an instance $< G = (V, E) >$ of $gap - VC_3$, denote $|V| = n$, $|E| = m$, $m_1 = \frac{144}{284} \cdot n$ and $m_2 = \frac{145}{284} \cdot n$. We construct an instance $< S, L >$ of ML such that S is a set of $m + 1$ strings, each string of length $k = n$, and $L = -(m + \frac{m_1+m_2}{2}) \cdot \log n$.

The first string in S consists of all zeros (the all zeros string), i.e.,

$$\underbrace{00...0...00}_{k}$$

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

$$\underbrace{\overbrace{00..00}^{i-1}1\overbrace{00..00}^{j-i-1}1\overbrace{00..00}^{k-j}}_{k}$$

where only the i-th and the j-th positions are set to 1. These m strings are called "edge strings". From now on, the trees we refer to have leaves with labels generated by this construction.

We use asymptotic properties of likelihood of trees, so most claims will hold when the input graph is large enough (*i.e.* $n = |V|$ is large enough). In our proof, we deal with small size subtrees or subforests, containing at most $2 \cdot \log\log n$ leaves. We will need the following relation for the expressions in the likelihood of the subforests to hold (see corollary 3): $\lim_{n\to\infty}[C^d_{S\cup z,T}/(\log(k_c) \cdot pars(S\cup z, T))] = 0$ and $\lim_{n\to\infty}[C^u_{S\cup z,T}/(\log(k_c) \cdot pars(S\cup z, T))] = 0$. According to our reduction the parsimony score (and k_{nc}, the number of non-constant sites) of such subtrees and subforests is no more than $4 \cdot \log\log n$. So according to Theorem 4.7 and corollaries 2, 3, and 4, it is enough that $k_c = k - k_{nc}$ will be doubly exponential in these parameters (the size and the parsimony of these subforests) to get these relations.

The proof strongly relies on quantitative relations between parsimony and likelihood as proved in [21].

5.2 Likelihood of Canonical Trees

In this section we show that for every $\varepsilon > 0$ there is an $n_0 > 0$ such that for $n > n_0$, the ratio between the log likelihood and the maximum log likelihood of some canonical tree is upper bounded by $(1 + \varepsilon)$.

Given an ML tree, T, if it is in canonical form, we are done. Otherwise we locate subtrees of T, $T_1, T_2, ..., T_\ell$ with a common root, such that the number of leaves in $\bigcup_{i=1}^{l} T_i$ is in the interval $[\log\log n, 2 \cdot \log\log n]$. Notice that this is a subforest as there may be other subtrees rooted at the same node. It is easy show that such a subforest always exists (lemma 5.4). On the next step we show that the ratio of the log-likelihood of such subforest when, the all zero labelling is placed in its root, and the log-likelihood of the same subforest with any other labellings in its root, is small.

Lemma 5.4 *Suppose T is a rooted tree and v is an internal node such that the number of leaves below v is at least q. Then v has a descendent, u, such that u has a forest consisting of ℓ subtrees $T_1, T_2, ..., T_\ell$ ($\ell \geq 1$) rooted at u, and the number of leaves in the forest $\bigcup_{i=1}^{\ell} T_i$ is in the interval $[q, 2 \cdot q]$.*

Lemma 5.5 *Let h be the root of a subforest $F \subseteq T$. Let u be an internal node in F ($u \neq h$), whose degree is $r \geq 9$, and let $s \in \{0, 1\}^k$. Consider an assignment of labels to internal nodes of F, where h is assigned s. Among such assignments, those that optimize the parsimony score label u with 0^k.*

Proof. It suffices to prove the claim for every position separately. The internal node u have $r-1$ subtrees below it, and one edge "above" it, leading to h. Out of these subtrees, at most three have "1" in the position of interest (by the construction of edges string and the fact that our graphs are of degree 3). For the other $r-4>4$ subtrees, since their leaves have 0 in the position, the most parsimonious assignment will label all their nodes with 0 (this can be seen by, for example, running Fitch algorithm [11]). Therefore u has at least 5 neighbour nodes with 0, and at most 4 with 1. Any parsimonious assignment will thus label u with 0. ■

The proof of the following Lemma is similar to that of Lemma 5.5.

Lemma 5.6 *Let h be the root of a subforest $F \subseteq T$. Suppose the degree of h is $r \geq 9$. Consider the parsimony score of F when h is assigned $s \in \{0,1\}^k$, and when h is assigned 0^k. The latter score is better (smaller).*

Lemma 5.7 *Let h be the root of a subforest F (h has at least two children in F). Suppose that in each position, the leaves labelled with "1" are at distance at least 4 from h. Then the max parsimony score on F is achieved with the all zero labelling in h.*

Proof. Consider an arbitrary position. There are at most three leaves x,y,z with '1' in this position. Let $LCA(x,y)$, $LCA(x,y,z)$ denote the least common ancestors of x,y and x,y,z respectively (see figure 5). Suppose, without loss of generality, that $LCA(x,y)$ is equal to $LCA(x,y,z)$ or is below it in F. For any node j, we denote by $pa(j)$ the parent of j. Consider three cases:

1. $h = LCA(x,y,z)$ and $LCA(x,y) \neq LCA(x,y,z)$:

   By Fitch algorithm the best assignment to the nodes in the path from z to h is '0'. Thus if we assign '0' to h we may lose 1 in the score due to the node just below h in the path between $LCA(x,y)$ and h, but lose nothing in the score due to the edges to the other children of h. On the other hand, if we assign '1' to h we lose 1 in the score due to the pre-last node in the path from z to h, and may lose 1 in the score due to the node just below h in the path between $LCA(x,y)$ and h. Thus the '0' labelling to h is not worse than the '1' labelling.

2. $h = LCA(x,y,z)$ and $LCA(x,y) = LCA(x,y,z)$: By Fitch algorithm the best assignment to the nodes in the paths from z, y, and x to h is '0'. If we assign '1' to h we loose 1 on each edge leading to h, a total loss of 3. If we assign '0' to h we lose nothing on the adjacent to h.

3. $h \neq LCA(x,y,z)$:

   Since h has at least two children, in this case all the leaves under one of these children are '0', the algorithm of Fitch assign '0' to this node. Since $LCA(x,y,z)$ is below h, Fitch's algorithm will assign "1" to at most one of h children. Thus the '0' labelling to h is not worse than the '1' labelling.

■

Corollary 5 *Let h be a root of a subforest F. Suppose all leaves having "1" in the position are either at distance ≥ 4 from the root, or have an internal node of degree ≥ 9 in the path to the root. Then the parsimony score of F when labelling the root h with 0 at this position is at least as good as when labelling the root with 1.*

Proof. According to Lemma 5.5 if there is an internal node of degree ≥ 9, it is better to assign 0 to this node. This enables us to disregard the "1" leaves of distance less than 4 from h with high degree node on their path to h. We can now apply Lemma 5.7. ■

Theorem 5.8 *Let T be a tree whose leaves are labelled by a subset of the edge strings. Let F be a subforest of T, rooted at h, and let $s \in \{0,1\}^k$ be a label of h. The parsimony score of F with the 0^k label at the root is worse by less than 8^4 than the parsimony score of F with label s at its root.*

Proof. If the degree of h is larger than 8, then by Lemma 5.5 the best assignment to h is 0^k and we are done. We say that a leaf in F is *dangerous* if its distance from h less than 4, and all its ancestors have degree ≤ 8. The number of dengerous leaves is smaller than $(8 + 8^2 + 8^3)$. Every dangerous leaf has 2 positions where it is labelled "1". Each such position can be "1" in at most 3 leaves because G is of degree 3. Therefore for any of these positions, changing the label at h from 1 to 0 will worsen the parsimony score by at most 3. There are at most $2 \cdot (8+8^2+8^3)$ such positions. So changing to 0 at h will cause at most $2 \cdot 3 \cdot (8+8^2+8^3) < 8^4$ parsimony degradation. According to Lemma 5, in all other positions, the "0" label at h is optimal. ■

The following Lemma was proved by Day *at. al.* [5, 1].

Lemma 5.9 *Let $S' \subseteq S$ be a subset of the "reduction strings", which contains the all zero string. The structure of the best parsimony tree for S' is canonical.*

Theorem 5.10 *For every $\varepsilon > 0$ there is an n_0 such that for all $n \geq n_0$, if $S \subseteq 0, 1^n$ is a set of $m+1$ reduction strings on a graph with n nodes, the following hold: Let T_{ML} denote an ML tree for S, and let p^*_{ML} and be an optimal edges length for this ML tree. Then there is a canonical tree for S, T_{Ca}, with optimal edges length p^*_{Ca}, such that:*

$$\frac{\log(P(S|p^*_{ML}, T_{ML}))}{\log(P(S|p^*_{Ca}, T_{Ca}))} > (1-\varepsilon) \ .$$

Proof. We start from any ML tree, T_{ML}, and show how to transform it to a canonical tree, T_{Ca}, with "close enough" log likelihood, in a sequence of up to $n/\log\log(n)$ steps. Each step involves a small, local change to the current tree: We identifying a subforest with a common root and number of leaves in the interval $[\log\log(n), 2 \cdot \log\log(n)]$. By Lemma 5.4, if the root of the whole tree has a subtree with more than $\log\log(n)$ leaves, we can find such a subforest. In such case, we first uproot this subforest and move it to the root. By Theorem 5.8 the parsimony score of such subforest with the 0^k label at the root is worse by less than $B \equiv 8^4$ than the parsimony score of F with any $s \in \{0,1\}^k$ label at its root. By lemma 4.6 we can assume s have zero in each position all the subforest's leaves have zero. Since the number of leaves in F is at most $2\log\log(n)$, the number of *constant sites* k_c, is at least $n - 4\log\log(n)$, so $\log(k_c) = \log(n) - o(\log n)$. Applying corollary 3 to $l_h = 0^k$ and $l_h = s \in \{0,1\}^k$

$$2^{-log(k_c)\cdot pars(S\cup 0^k,F) - C^d_{S\cup 0^k,F}} \leq Pr(S, l_h = 0^k | p^*, F) \leq 2^{-log(k_c)\cdot pars(S\cup 0,F) - C^u_{S\cup 0^k,F}}.$$

$$2^{-log(k_c)\cdot pars(S\cup s,F) - C^d_{S\cup s,F}} \leq Pr(S, l_h = s | p^*, F) \leq 2^{-log(k_c)\cdot pars(S\cup s,F) - C^u_{S\cup s,F}}.$$

The parsimony score on such F with $l_h = 0^k$ at its root is no more than the number of "1" entries, which is bounded by $4\log\log(n)$. The size of F (the number of vertices) is at most $4\log\log(n)$. The function $C^u_{S\bigcup z,F}$ is a positive, sub-quadratic function of F's size and the parsimony score. Thus, are $C^u_{S\cup 0^k,F} = o((\log\log(n))^2)$.

$$\begin{aligned} & \log Pr(S, l_h = s|p^*, F) - \log Pr(S, l_h = 0^k|p^*, F) \\ \leq \ & -\log(k_c)\cdot pars(S\cup s, F) - C^u_{S\cup s,F} - (-\log(k_c)\cdot pars(S\cup 0^k, F) - C^d_{S\cup 0^k,F}) \\ = \ & -\log(k_c)\cdot (pars(S\cup s, F) - pars(S\cup 0^k, F)) + C^u_{S\cup 0^k,F} - C^d_{S\cup s,F} \end{aligned}$$

$$
\begin{aligned}
&\leq B\log(k_c) + C^u_{S\cup 0^k,F} - C^d_{S\cup s,F} \\
&\leq B\log(k_c) + O(\log^2\log(n)) \\
&= B\log(n) + o(\log n)
\end{aligned}
$$

Therefore, according to Theorem 4.4, when moving this forest to the root, the total log likelihood of the tree decreases by less then $B\log(n) + o(\log n)$. According to Lemma 5.9 we can rearrange such a subforest with the all zero root in a canonical form such that its parsimony score will not become worse, let F_c denote such such canonical rearrangement. Thus by corollary 3 and corollary 4:

$$
\begin{aligned}
&\log Pr(S, l_h = 0^k | p^*, F) - \log Pr(S, l_h = 0^k | p^*, F_c) \\
\leq\ & -\log(k_c)\cdot pars(S\cup 0^k, F) - C^u_{S\cup 0^k,F} - (-\log(k_c)\cdot pars(S\cup 0^k, F_c) - C^d_{S\cup 0^k,F_c}) \\
\leq\ & C^u_{S\cup 0^k,F} - C^d_{S\cup s,F_c} \\
\leq\ & O(\log^2\log(n)) \\
=\ & o(\log n)
\end{aligned}
$$

Therefore, such rearrangement can decrease the log likelihood of the tree by less $O((\log\log(n))^2) = o(\log n)$. If we reached a situation where all subtrees are smaller than $\log\log n$, we rearrange each subforest of size in the range $[\log\log(n), 2\cdot\log\log(n)]$ separately. According to Theorem 4.10, the log-likelihood of all canonical trees is larger than $-n\log(n)$. We just showed the existence of a canonical tree whose log likelihood differs from the log likelihood of any ML tree by less than $Bn\log(n)/\log\log(n)$ (for large enough n). Thus there must be a constant $K > 0$ such that the log-likelihood of any ML tree is at most $-K\cdot n\log(n)$, and consequently there is a canonical tree such that the ratio between the log likelihood of the ML tree and this tree is

$$
\frac{-K\cdot n\log(n)}{-K\cdot n\log(n) - O(n\cdot\log n/\log\log(n))} = 1 + O(\frac{1}{\log\log n}) < 1+\varepsilon.
$$

■

5.3 Correctness of the Reduction

In this section we complete our proof by showing that indeed we have a reduction from $GAP{-}VC_3$ to ML. The basic idea is to show that if G has a small enough cover, then the likelihood of the corresponding canonical tree is high (this is the easy direction), and if the likelihood is high, then there is a small cover (the harder direction). The translation of sizes, from covers to log likelihood, and vice versa, is not sharp but introduces some slack. This is why a hard approximate version of vertex cover is required as our starting point.

The next Lemma establishes a connection between MP and VC, and was used in the NP-hardness proof of MP.

Lemma 5.11 *[5, 1] $G = (V, E)$ has a vertex cover of size c if and only if there is a canonical tree with parsimony score $c + m$, where c is the degree of the root.*

Theorem 5.12 *For every $0 < \varepsilon$ there is an n_0 such that for $n \geq n_0$, given a degree 3 graph $G = (V, E)$ on n nodes and m edges, with a cover of size at most c, the following holds: There is a tree T such that the log - likelihood of the tree satisfies*

$$
\log(Pr(S|p^*(S,T),T)) \geq -(1+\varepsilon)\cdot(m+c)\cdot\log n.
$$

On the other hand, if the the size of every cover is $\geq c$ then the log likelihood of each tree, T, satisfies

$$\log(Pr(S|p^*(S,T),T)) \leq -(1-\varepsilon)\cdot(m+c)\cdot\log n.$$

Proof. Suppose G has a vertex cover of size $\leq c$. Since G's is of bounded degree 3: $m/3 \leq c \leq m$ and $n \leq m \leq 1.5\cdot n$. According to Lemma 5.11, there is a canonical tree, T, with parsimony score $c+m$, such that the degree of its root is c. According to Theorem 4.10, this tree has log likelihood $-(c+m)\cdot\log(n)+\theta(n)$. Since $m, c = \theta(n)$, we have $n = o((c+m)\cdot\log(n))$, so $\log(Pr(S|p^*(S,T),T)) = -(c+m)\cdot\log(n)+\theta(n)$ implies tat for every $\varepsilon > 0$ and large enough n: $\log(Pr(S|p^*(S,T),T)) > -(m+c)\cdot\log(n)\cdot(1+\varepsilon)$.

For the other direction, suppose the size of every cover of G is $\geq c$. According to Lemma 5.11 the parsimony score of each canonical tree is at least $c+m$. Thus the likelihood of each canonical tree is at most $-(c+m)\cdot\log(n)+C$ where $C \leq m\cdot\max\{C_1^u, C_2^u, C_3^u\} = \theta(n)$. Since $m, c = \theta(n)$ we get that the likelihood of each canonical tree is at most $-(c+m)\cdot\log(n)+\theta(n) \leq -(c+m)\cdot\log(n)\cdot(1-\varepsilon_1)$ according to Theorem 5.10 the likelihood of the best tree is $\leq -(c+m)\cdot\log(n)\cdot(1-\varepsilon_1)(1-\varepsilon_2)$ where $\varepsilon_1, \varepsilon_2$ are arbitrarily small, thus for every ε there is n_0 such that for $n > n_0$ the likelihood of the best tree is $\leq -(c+m)\cdot\log(n)\cdot(1-\varepsilon)$. ■

Theorem 5.13 *ML is NP-hard.*

Proof. Let $1+\varepsilon_c = 1.0069 = \frac{m_2}{m_1}$, let c be the size of the minimal cover in G. Suppose $c < m_1$, then according to Theorem 5.12 there is a tree whose likelihood is at least $-(1+\varepsilon)\cdot(m+c)\cdot\log n$, where ε is arbitrarily small. Since $\frac{m_1+m_2}{2} = m_1\cdot(1+\frac{\varepsilon_c}{2})$, and $\frac{m}{3} \leq m_1 \leq m$ (the degree of the graph is at most 3). For small enough ε we get $L = -(\frac{m_1+m_2}{2}+m)\cdot\log n = -(m_1\cdot(1+\frac{\varepsilon_c}{2})+m)\cdot\log n \leq -(m+m_1)\cdot\log n\cdot(1+\varepsilon) \leq -(m+c)\cdot\log n\cdot(1+\varepsilon)$. Thus $(S,L) \in ML$.

Suppose every cover of G is larger than $c > m_2$, according to Theorem 5.12 the log likelihood of each tree is less than $-(1-\varepsilon)\cdot(m+c)\cdot\log n$, where ε is arbitrarily small. Since $\frac{m_1+m_2}{2} = m_2\cdot(1+\frac{1}{\varepsilon_c})$, $c > m_2$ and $\frac{m}{3} \leq m_2, m_1 \leq m$. For small enough ε we get

$$-(1-\varepsilon)\cdot(c+m)\cdot\log n \leq -(1-\varepsilon)\cdot(m_2+m)\cdot\log n \leq$$

$$-(1-\varepsilon)\cdot((\frac{m_1+m_2}{2}+m)\cdot\frac{1+\frac{1}{\varepsilon_c}}{4})\cdot\log(n) \leq -(m+\frac{m_1+m_2}{2})\log(n) = L$$

. Thus $(S,L) \notin ML$. ■

6 Other Models for Maximum Likelihood

In this section we prove NP-hardness of maximum likelihood under Jukes-Cantor model [14]. This model is a special case of Kimura and other models of evolution in DNA, thus ML is NP-hard for DNA sequences. If we use similar models for proteins we can conclude ML is hard for protein sequences.

Let α denote a parameter, suppose we have a c state alphabet (*e.g.* for DNA sequences $c = 4$), the probability that a nucleotide will not change after time t is:

$$1-p_e = \frac{1}{c}\cdot(1+(c-1)\cdot e^{-c\cdot\alpha\cdot t}).$$

The probability that a nucleotide will change to another nucleotide after time t is

$$p_e = \frac{c-1}{c} \cdot (1 - e^{-c \cdot \alpha \cdot t}).$$

The likelihood of a tree under this model is defined in a way similar (but not the same) to equation 2:

$$L(S|T, \mathbf{p}) = \sum_{\lambda \in \{0,1,..,c-1\}^{k \times r}} \prod_{e \in E(T)} \frac{p_e}{c-1}^{d_e(\lambda)} \cdot (1 - p_e(\lambda))^{k - d_e(\lambda)} \quad (3)$$

According to [21] we get for this model relation that are similar to the theorems, lemmata and corollaries in section 4 (with different $C^u_{T,pars(S,T)}$ and $C^d_{T,pars(S,T)}$ that have the same order). Thus the same reduction holds for this model.

7 Conclusions and Further Research

In this work, we proved that ML reconstruction of phylogenetic trees is computationally intractable. We used the simplest model of substitution - Neyman two states model [17]. Furthermore, we generalised our NP-hardness proof to the Jukes-Cantor model [14]. This model is a special case of Kimura and other models of DNA substitution.

While resolving a 20 year old problem, this work raises a number of additional ones. Our proof techniques can be extended to show that there is a constant $\delta > 0$ such that finding a tree whose *log likelihood* is within $(1+\delta)$ of the optimum is hard. Can such inapproximability be proved with respect to the *likelihood* itself? Vertex cover, which is the starting point for the reduction, has a simple 2-approximation result. What about approximation algorithms for log likelihood? What can be said about the complexity of ML when restricted to trees under a *molecular clock*?

And finally, it would be nice to identify regions where ML is *tractable*. However, it is not even known what is the complexity of *small ML*, where the sequences and the unweighted tree are given, and the goal is to find optimal edge lengths. In practice, local search techniques such as EM or hill climbing seem to perform well, but no proof of performance is known, and multiple maxima [19, 3] shed doubts even on the (worst case) correctness of this approach.

Acknowledgements

We wish to thank Isaac Elias for helpful discussions, and Sagi Snir for reading early drafts of the manuscript.

References

[1] L. Addario-Berry, B. Chor, M. Hallett, J. Lagergren, A. Panconesi, and T. Wareham. Ancestral maximum likelihood of evolutionary trees is hard. *Jour. of Bioinformatics and Comp. Biology*, 2(2):257–271, 2004.

[2] P. Berman and M. Karpinski. On some tighter inapproximability results. *Proc. 26th ICALP*, 1999.

[3] B. Chor, M. D. Hendy, B. R. Holland, and D. Penny. Multiple maxima of likelihood in phylogenetic trees: An analytic approach. *Mol. Biol. Evol.*, 17(10):1529–1541, 2000.

[4] W. Day. The computational complexity of inferring phylogenies from dissimilarity matrix. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.

[5] W. Day, D. Johnson, and D. Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, 81:33–42, 1986.

[6] W. Day and D. Sankoff. The computational complexity of inferring phylogenies by compatibility. *Systematic Zoology*, 35(2):224–229, 1986.

[7] I. Dinur and S. Safra. On the importance of being biased (1.36 hardness of approximating vertex-cover). *Annals of Mathematics (accepted)*, 2005.

[8] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

[9] J. Felsenstein. Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Meth. in. Enzym.*, 266:419–427, 1996.

[10] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts, 2004.

[11] W. M. Fitch. Toward defining the course of evolution: minimum change for specified tree topology. *Systematic Zoology*, 20:406–416, 1971.

[12] L. Foulds and R. Graham. The steiner problem in phylogeny is np-complete. *Advances in Applied Mathematics*, 3:43–49, 1982.

[13] J. Hastad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001.

[14] T. H. Jukes and C. R. Cantor. Evolution of protein molecules. *In H. N. Munro, editor, Mammalian protein metabolism*, pages 21–132, 1969.

[15] M. Karpinski. Approximating bounded degree instances of np-hard problems. *FCT*, 2001.

[16] M. Koshi and R. Goldstein. Probabilistic reconstruction of ancestral nucleotide and amino acid sequences. *Journal of Molecular Evolution*, 42:313–320, 1996.

[17] J. Neyman. Molecular studies of evolution: A source of novel statistical problems. *In S. Gupta and Y. Jackel, editors, Statistical Decision Theory and Related Topics*, pages 1–27., 1971. Academic Press, New York.

[18] M. steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of classification*, 9:71–90, 1992.

[19] M. Steel. The maximum likelihood point for a phlogenetic tree is not unique. *Syst. Biol.*, 43:560–564, 1994.

[20] M. Steel and D. Penny. Parsimony, likelihood and the role of models in molecular phylogenetics. *Mol. Biol. Evol.*, 17:839–850, 2000.

[21] C. Tuffley and M. Steel. Link between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bulletin of Mathematical Biology*, 59(3):581–607, 1997.

[22] T. Wareham. On the computational complexity of inferring evolutional trees. *Technical Report 93-01, Department of computer science, Memorial University of Newfoundland*, 1993.

[23] Z. Yang, S. Kumar, and M. Nei. A new method of inferring of ancestral nucleotide and amino acid sequences. *Genetics*, 141:1641–1650, 1995.

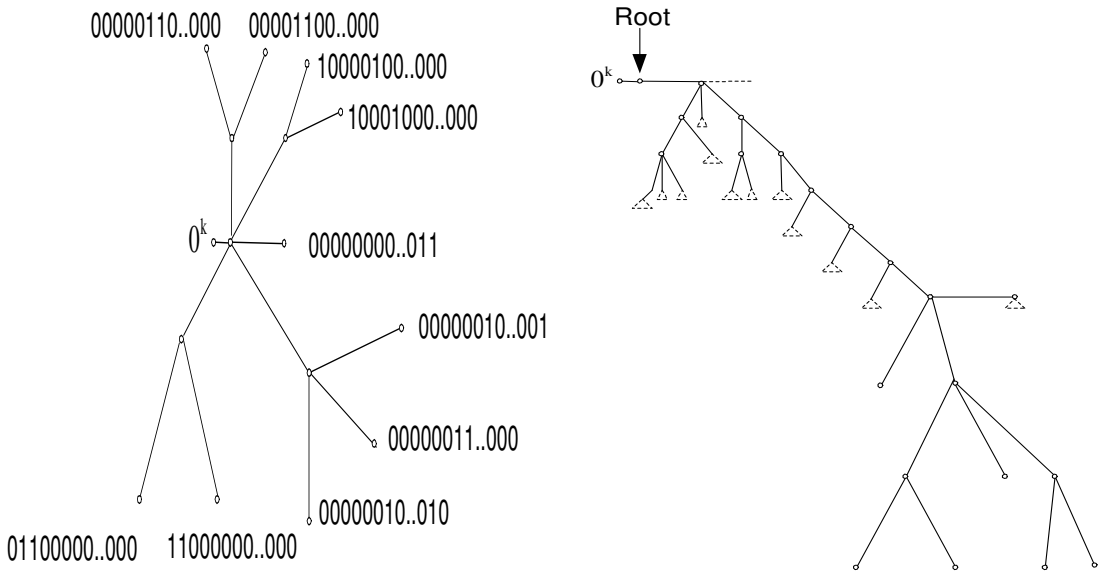

Figure 1: Canonical (top) and non-canonical (bottom) trees.

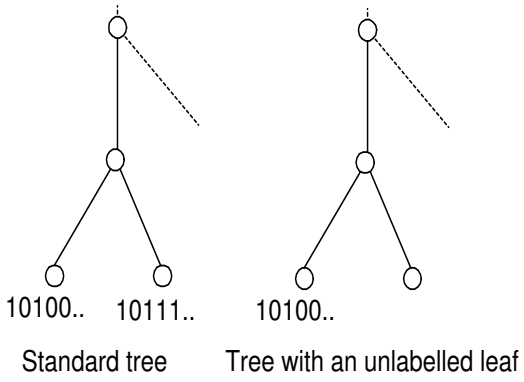

Figure 2: Regular graph and graph with unlabelled leaf.

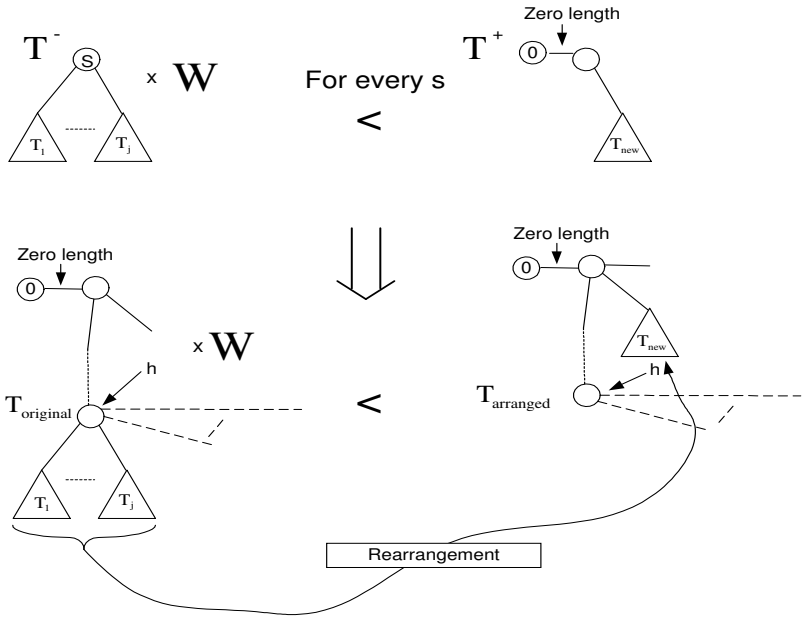

Figure 3: Theorem 4.4.

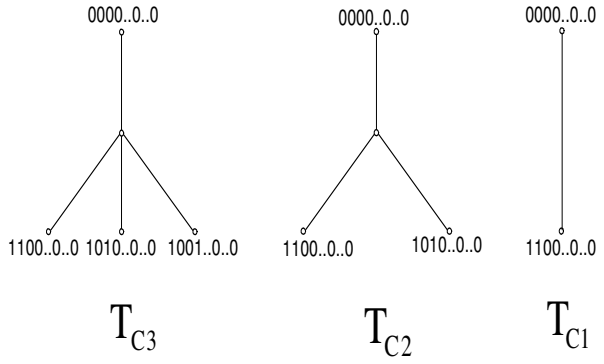

Figure 4: Building blocks of the maximum likelihood tree for our reduction.

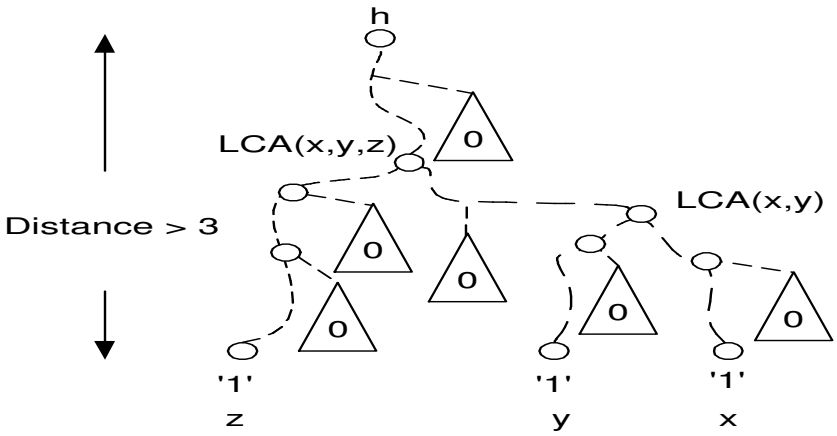

Figure 5: Lemma 5.7.