



Maximum likelihood of evolutionary trees: hardness and approximation

Benny Chor and Tamir Tuller*

School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel

Received on January 15, 2004; accepted on March 27, 2005

ABSTRACT

Motivation: Maximum likelihood (ML) is an increasingly popular optimality criterion for selecting evolutionary trees. Yet the computational complexity of ML was open for over 20 years, and only recently resolved by the authors for the Jukes–Cantor model of substitution and its generalizations. It was proved that reconstructing the ML tree is computationally intractable (NP-hard). In this work we explore three directions, which extend that result.

Results: (1) We show that ML under the assumption of molecular clock is still computationally intractable (NP-hard). (2) We show that not only is it computationally intractable to find the exact ML tree, even approximating the logarithm of the ML for any multiplicative factor smaller than 1.00175 is computationally intractable. (3) We develop an algorithm for approximating log-likelihood under the condition that the input sequences are *sparse*. It employs any approximation algorithm for *parsimony*, and asymptotically achieves the same approximation ratio. We note that ML reconstruction for sparse inputs is still hard under this condition, and furthermore many real datasets satisfy it.

Contact: tamirtul@post.tau.ac.il

1 INTRODUCTION

Understanding the origin and evolution of extant and extinct species is a fundamental scientific quest. Today, phylogenetic trees are widely used as the accepted evolutionary model, and are mostly based upon molecular sequences (amino acid or DNA) data. The space of candidate trees grows exponentially with the number of species, implying that even on modern computers an exhaustive search over all trees is infeasible (except for few species, no more than ~ 20).

Two frequently used reconstruction criteria are maximum parsimony (MP) and maximum likelihood (ML). But are they solvable in a computationally efficient manner? MP was proved intractable almost 20 years ago (Day *et al.*, 1986). The analogue question for ML remained unsolved. This created a strange situation, because most practitioners believe that ML is computationally harder than MP. Namely,

computer programs running on the same sets of sequences tend to take much longer to solve ML than MP. Yet ML remained ‘unclassified’, leaving the existence of an efficient ML algorithm a possibility.

In Chor and Tuller (2005), we resolved this question (the full version is available from <http://www.cs.tau.ac.il/~bchor/> and <http://www.cs.tau.ac.il/~tamirtul/>). We showed that ML on phylogenetic trees is indeed computationally intractable, or NP-hard. Important ingredients in that work are the quantitative relations between MP and ML, which were investigated in Tuffely and Steel (1997). These relations are also used in the present work.

In this paper we explore three extensions of the intractability result. First, we show that even when restricting the trees to satisfy the molecular clock assumption, ML reconstruction remains computationally hard. Even though molecular clock is violated for general phylogenetic trees (Goodman, 1981), it seems to hold for cases of closely related species (Yoder and Yang, 2000; Gaunt and Miles, 2002; Margoliash, 1963; Zuckerkandl and Pauling, 1962, 1965). Furthermore, it was widely used in reconstruction of phylogenetic trees (e.g. Nei *et al.*, 2000; Graur and Li, 1999). This stronger hardness result sheds further light on the intractability of ML, which remains invariant under certain restrictions, such as molecular clock.

We then investigate the hardness of ML approximation, and show it is NP-hard to approximate log-ML within any factor $1 + \varepsilon$ where $\varepsilon < 0.00175$. The value $\varepsilon = 0.00175$ is indeed rather small, but this preliminary result suffices to show that ML reconstruction does not have a polynomial time approximation scheme (Papadimitriou, 1993). It is a challenging task to find larger ε . As an upper bound, in contrast to parsimony, where a factor 2 approximation is known, no constant factor approximation algorithm for log ML has been found so far.

Finally, we develop an approximation algorithm for log ML under a sparseness condition on the problem’s input. Given an A approximation algorithm for parsimony, our algorithm achieves the same A approximating ratio for log-ML asymptotically (for long enough input sequences). We note, however, that ML reconstruction is still computationally hard even for sparse inputs. To the best of our knowledge, this algorithm

*To whom correspondence should be addressed.

is the first such with provable performance guarantees for approximating log-likelihood under any reasonable, general restriction. We demonstrate a few results of running our algorithm on synthetic and real datasets.

2 MODEL, DEFINITIONS AND NOTATIONS

In this section we describe the model and basic definitions that we will use later.

DEFINITION 2.1. [*Phylogenetic trees and characters* (Tuffely and Steel, 1997)] *A phylogenetic tree on n leaves is a tree $T = (V(T), E(T))$ having no vertices of degree two, and such that each leaf (degree one vertex) is given a unique label from $[n] = \{1, \dots, n\}$. For convenience, we identify each leaf with its label. A non-leaf vertex is called an internal vertex. A function $\lambda : [n] \rightarrow \{0, 1\}$ is called a state function for T ; if $\hat{\lambda} : V(T) \mapsto \{0, 1\}$ is such that $\hat{\lambda}$ agrees with λ on the leaves of T , then $\hat{\lambda}$ is called an extension of λ (on T). In a similar way we define the functions $\lambda^k : [n] \mapsto \{0, 1\}^k$ and $\hat{\lambda}^k : V(T) \mapsto \{0, 1\}^k$. This last function is called a labelling function for T . If $\hat{\lambda}^k(v) = s$, we say that the s is the label of vertex v .*

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

DEFINITION 2.2. [*MP score*] *Let $S = [s(1), s(2), \dots, s(n)] \in \{0, 1\}^{n \times k}$ be a set containing n binary strings of length k . Let T be a binary tree with n leaves. Let $\hat{\lambda}^k : V(T) \mapsto \{0, 1\}^k$ be a labelling function for T that agrees with S on the leaves. The parsimony score for T and the labelling, $\text{pars}(S, \hat{\lambda}^k, T)$, is the value of $\sum_{e \in E(T)} d_e(\hat{\lambda}^k)$ for this labelling. The MP score for S is $\text{pars}(S) = \min_{\lambda^k, T} \text{pars}(S, \lambda^k, T)$. The ‘MP tree (s)’ for the set S is a tree (or trees) and labelling that attain this minimum score.*

In the likelihood setting, the basic model we use is the Neyman (1971) two-states model. For a tree T , let $\mathbf{p} = [p_e]_{e \in E(T)}$ ($0 \leq p_e \leq 1/2$) be the edge probabilities vector. According to the model:

- The probability of a net change of state (from ‘1’ to ‘0’ or vice versa) occurring across the edge e (a ‘mutation event’) equals p_e (the ‘edge mutation probability’ of e).
- Mutation events on different tree edges are independent.
- Different sites mutate independently.

Let $S = [s(1), s(2), \dots, s(n)] \in \{0, 1\}^{n \times k}$ be the observed (given) sequences of length k over n taxa (n leaves). The likelihood, $L(S|T, \mathbf{p})$, of observing such S , given the tree T with $r \leq n - 2$ internal nodes and the edge probabilities vector

\mathbf{p} is defined as

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

where \mathbf{a} ranges over all combinations of assigning characters states (0 or 1) to the r internal nodes of T . This notion of ML is termed maximum average likelihood in Steel and Penny (2000). 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(s) 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(s) 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, Felsenstein, 1981; Steel, 1994; Tuffely and Steel, 1997, for more details). It is easy to see that by site independence, an equivalent way to define the likelihood of observing S in the weighted tree T is:

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

For the hardness results for ML, our reductions are from a ‘gap’ version of vertex cover problem on specific graphs (degree three graphs) (Berman and Karpinski, 1999). Given an input to the vertex cover problem, $G = (V, E)$, we generate a set of string, S , as input to our ML problem. The first string in S consists of the all zeros string, that is,

$$\underbrace{00\dots0\dots00}_k$$

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

$$\underbrace{\underbrace{00\dots00}_{i-1} 1 \underbrace{00\dots00}_{j-i-1} 1 \underbrace{00\dots00}_{k-j}}_k$$

where only the i -th and the j -th characters are set to 1. We call these strings ‘edge strings’. We point out that the transformation itself (from the graph G to this set of sequences) is identical to the one for MP by (Day *et al.*, 1986).

In Chor and Tuller (2005) we show that if L is the likelihood of the ML 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 smallest cover, then as $n \rightarrow \infty$,

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

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

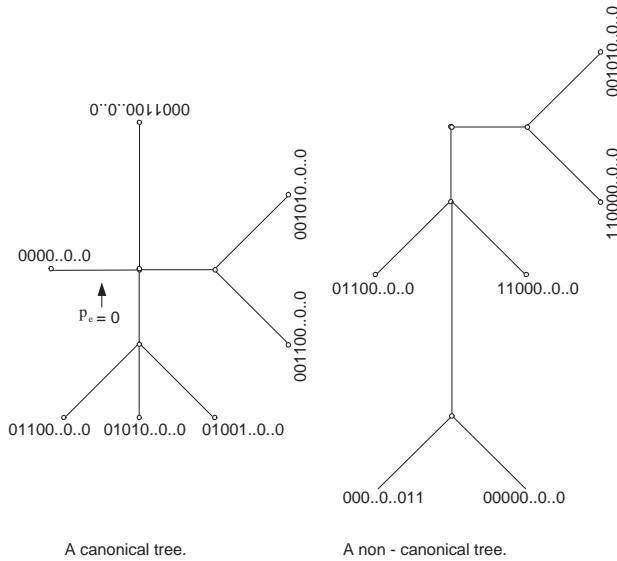


Fig. 1. Canonical and non-canonical trees.

Canonical trees, or variants of them, play an important role in the proof of this relationship. We say that a tree has a canonical form if the following properties hold (see Fig. 1):

DEFINITION 2.3.

- 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 edge-probability of the edge going to this leaf is 0.*
- 2 *All leaves are at depth either 1 or 2 from the root.*
- 3 *If a leaf is at depth 2 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. We denote subtrees with 1, 2, or 3 leaves by T_{C_1} , T_{C_2} , and T_{C_3} , respectively.*

We will also deal with the trees T_{C_1} , T_{C_2} , and T_{C_3} in isolation, detached from a canonical tree. In this case, the detached node is labelled by the all 0 vector, in accordance with the situation when it is connected to the canonical tree.

3 RESULTS

3.1 Molecular clock

Let p_e denote the ‘edge mutation probability’ of an edge, e . In a standard Poisson model, $q_e = -\frac{1}{2} \ln(1 - 2 \cdot p_e)$ is the expected number of mutations along an edge e . Unlike the p_e s, the q_e s terms are additive (along paths in the tree), so q_e is usually taken as the ‘edge length’ of e (Hendy and Penny, 1993). The transformation $p_e \rightleftharpoons q_e$ is invertible, as $p_e = \frac{1}{2} \cdot (1 - e^{-2 \cdot q_e})$. The probability that the character states differ at the endpoints of a path Π in a tree, equals $p_\Pi = \frac{1}{2} \cdot (1 - e^{-2 \cdot q_\Pi})$, where q_Π is the sum of q_e values of the edges

in Π (Hendy and Penny, 1993). Under the molecular clock assumption, trees contain an internal node (‘root’) such that the distance from each leaf to the root is the same. Let

$$q_1 = -\frac{1}{2} \cdot \ln(1 - 2 \cdot p) \Big|_{p=\frac{2}{n}} = -\frac{1}{2} \cdot \ln\left(1 - \frac{4}{n}\right),$$

and $q_{2,3} = q_1/2 = -\ln(1 - 2 \cdot p_{2,3})/2$.

Our molecular clock trees will have distance q_1 from the root to each of the leaves. The edges are of length q_1 or $q_{2,3}$. The following lemma shows that for n large enough, $p_{2,3}$ is arbitrarily close to $1/n$.

LEMMA 3.1. $p_{2,3} = (1/n) + O(1/n^2)$.

PROOF. $p_{2,3} = \frac{1}{2} \cdot (1 - \sqrt{1 - (4/n)})$. Using Taylor series expansion, we get $\sqrt{1 - (4/n)} = 1 - (2/n) + O(1/n^2)$. Thus $p_{2,3} = (1/n) + O(1/n^2)$.

The likelihood of the best tree under molecular clock is upper bounded by the likelihood of the ML tree. By the results of Chor and Tuller (2005) and the relation above, we show that the likelihood of the best tree under the molecular clock is not too low, compared to the likelihood of the ML tree. We get the desired NP-hardness result for ML tree under molecular clock using these two relations.

In our proof, we modify a canonical tree to fit the molecular clock assumption. We start from an original, canonical tree that does not satisfy the molecular clock assumption. In the modification, the decrease in the log-likelihood is relatively small. For a canonical tree T_{Ca} , let T_{Ca}^1 denote a tree with the same topology as T_{Ca} , such that the length of all the edges in its T_{C_2} and T_{C_3} subtrees equals $q_{2,3}$, the edges’ lengths in its T_{C_1} subtrees are q_1 , and the length of the edge going from the root to the all zero leaves is q_1 (and not 0 as in Definition 2.3). Since the distances of all the leaves from the root in T_{Ca}^1 is q_1 , this tree satisfies the molecular clock condition. Let p_{Ca}^1 denote these edges probabilities, and let p_{Ca}^* denote the best edge probabilities for a canonical tree with the same topology as T_{Ca}^1 . We will show that for every ε there is an n_0 such that for $n > n_0$:

$$1 > \frac{\log(\Pr(S|T_{Ca}, p_{Ca}^*))}{\log(\Pr(S|T_{Ca}^1, p_{Ca}^1))} > (1 - \varepsilon).$$

In Chor and Tuller (2005) we showed that for n large enough, there is a canonical tree whose log-likelihood is close to the log-likelihood of the ML tree. Given such a canonical tree, we have two additional steps in our proof. At first, we change the length of the edge from the root to the all zero string to q_1 . Then we change the length of the edges in all T_{C_1} subtrees to q_1 , and in all T_{C_2} and T_{C_3} subtrees to $q_{2,3}$ (in these subtrees, paths from a leaf to the root have two edges). After these two changes, the distances of all the leaves from the root is q_1 . We will show that these two stages cause relatively small decrease in the log-likelihood. Let T_{Ca} be a canonical tree, let T_{Ca}^0 be

a tree identical to T_{Ca} , except the length of the edge from the root to the all zero leaf is changed to q_1 (instead of 0 as in T_{Ca}).

LEMMA 3.2. *For every $\varepsilon > 0$ there is an n_0 such that for $n > n_0$:*

$$\frac{\Pr(S|T_{Ca}^0, p_{Ca}^0)}{\Pr(S|T_{Ca}, p_{Ca}^*)} \geq \frac{1}{e^2} - \varepsilon.$$

PROOF. Let h denote the vertex that is the common root of T_{Ca}^0 and T_{Ca} . Then

$$\begin{aligned} & \Pr(S|T_{Ca}^0, p_{Ca}^0) \\ &= \sum_{s \in \{0,1\}^n} \Pr(S|h = s, T_{Ca}^0, p_{Ca}^0) \\ &= \sum_{s \in \{0,1\}^n} \Pr(0^n|h = s) \cdot \Pr(S \setminus \{0^n\} | h = s, T_{Ca}^0, p_{Ca}^0) \\ &> \Pr(0^n|h = s) \cdot \Pr(S \setminus \{0^n\} | h = 0^n, T_{Ca}^0, p_{Ca}^0) \\ &= \Pr(0^n|h = s) \cdot \Pr(S|p_{Ca}^*, T_{Ca}) \\ &= \left(1 - 2/n - O(1/n^2)\right)^n \cdot \Pr(S|p_{Ca}^*, T_{Ca}) \\ &= (e^{-2} - \varepsilon) \cdot \Pr(S|p_{Ca}^*, T_{Ca}). \end{aligned}$$

COROLLARY 3.3. *By Lemma 3.2*

$$\log(\Pr(S|p_{Ca}^0, T_{Ca}^0)) = \log(\Pr(S|p_{Ca}^*, T_{Ca})) + O(1)$$

LEMMA 3.4. *The optimal edge length for each T_{C_i} is $2/n$.*

PROOF. By direct calculations.

By combining Lemma 5 from Tuffely and Steel (1997), which describes a connection between the log-likelihood and the parsimony score of a tree, together with direct (though somewhat tedious) calculations, we get the following:

Let $T_{C_1}^{MC}$ denote a tree identical to T_{C_1} , but whose its edge length is q_1 . Let $T_{C_2}^{MC}$, $T_{C_3}^{MC}$ denote trees identical to T_{C_2} , T_{C_3} , respectively, but whose edges' length is $q_{2,3}$. Recall that all these trees have one leaf labelled 0^n . Let $ML_1^{MC}(n)$, $ML_2^{MC}(n)$, $ML_3^{MC}(n)$ denote the log-likelihood of these trees, respectively.

LEMMA 3.5. *There are constants $d_1, u_1, d_2, u_2, d_3, u_3$ (which do not depend on n) such that for all n :*

1. $-2 \cdot \log(n) + d_1 \leq ML_1^{MC}(n) \leq -2 \cdot \log(n) + u_1.$
2. $-3 \cdot \log(n) + d_2 \leq ML_2^{MC}(n) \leq -3 \cdot \log(n) + u_2.$
3. $-4 \cdot \log(n) + d_3 \leq ML_3^{MC}(n) \leq -4 \cdot \log(n) + u_3.$

THEOREM 3.6. *ML under molecular clock is NP-hard.*

PROOF. Let T_{Ca} be a canonical tree with degree d of the root. By Theorem 4.8 in Chor and Tuller (2005), the

log-likelihood of T_{Ca} is $-(m+d) \cdot \log(n) + \theta(n)$. The log-likelihood is the sum of log-likelihoods of the various subtrees. From Lemmas 3.2, 3.4, and 3.5, the log-likelihood of T_{Ca}^1 is $-(m+d) \cdot \log(n) + \theta(n)$. Thus for every ε there is n_0 such that for $n > n_0$

$$\frac{\log(\Pr(S|T_{Ca}, p_{Ca}^*))}{\log(\Pr(S|T_{Ca}^1, p_{Ca}^1))} > (1 - \varepsilon).$$

Let T_{ML} and p_{ML}^* denote the ML topology and edge probability, respectively, for our inputs, by theorem 5.9 in Chor and Tuller (2005) for every ε there is n_0 such that for $n > n_0$:

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

Thus Theorem 5.9 in Chor and Tuller (2005) holds even when replacing T_{Ca} by T_{Ca}^1 . We can use this last relation to show that ML for trees under molecular clock is hard.

3.2 Hardness of approximating ML

NP-hard problems greatly differ in their approximabilities (Ausiello *et al.*, 1998). In this section, we prove that it is NP-hard to approximate ML on phylogenetic trees to within any multiplicative factor smaller than 1.00175. Our starting point here is a gap vertex cover problem (Karpinski, 2001) that was also used in Chor and Tuller (2005). It is known that the following gap vertex cover problem, GAP – VC₃, is NP-hard (Karpinski, 2001): Given a degree 3 graph $G = (V, E)$, does it have a vertex cover smaller than $144n/284$, or is every vertex cover larger than $145n/284$, where $n = |V|, m = |E|$. (For vertex covers in the intermediate range there is no requirement.)

Using this VC gap, we derive a gap for log ML using the fact that if G has a cover of size c then the log-likelihood of the reduction strings is $(1 + o(1))(m + c) \log n$. For degree 3 graphs, $m \leq 3n/2$. Denoting by $[c_\ell, c_u]$ the gap interval, this yields a $[(1 + o(1))(m + c_\ell) \log n, (1 + o(1))(m + c_u) \log n]$ gap for log ML. Dividing the right-hand side by the left-hand side, the ratio is at least $(3/2 + 145/284)/(3/2 + 144/284) > 1.001754$, implying that approximating log-likelihood to within 1.001754 is hard.

DEFINITION 3.7. *Gap problem for log-likelihood, gap – log – ML[L_1, L_2]*

Input: A set of sequences, S , two negative numbers, L_1 and L_2 .

Question: Does S have a tree with log-likelihood larger than L_1 , or is the log-likelihood of each tree smaller than L_2 ? (In case the log ML is in the intermediate range, there is no requirement.)

Let n denote the length of the strings in the ML problem, let m denote the number of strings in the ML problem, let ε denote

an arbitrary small positive number. Let

$$L_d(\varepsilon) = -(1.5 \cdot n + 145n/284) \cdot \log n \cdot (1 - \varepsilon),$$

and let

$$L_u(\varepsilon) = -(1.5 \cdot n + 144n/284 \cdot n) \cdot \log n \cdot (1 + \varepsilon).$$

THEOREM 3.8. *For every $\varepsilon > 0$ the gap problem $GL - ML[L_d(\varepsilon), L_u(\varepsilon)]$ is NP-hard.*

PROOF. Given an instance $\langle G = (V, E) \rangle$ of gap- VC_3 , we use the same input set to our problem as in the reduction in Chor and Tuller (2005). By our construction, since G has degree 3, any cover is of size $\Theta(n) = \Theta(m)$. By Theorem 5.11 of Chor and Tuller (2005), if there is a vertex cover of size $|C|$ in G then the maximum log-likelihood of the tree for this ML instance is

$$-(|E| + |C|) \cdot \log n + O((n \cdot \log n)/(\log \log n)).$$

Thus for every $\varepsilon > 0$ there is an n_0 such that: For all $n > n_0$, if every vertex cover in G is of size $\geq (145/284)n$ then the log-likelihood of every tree in our reduction is

$$\leq -\left(1.5 \cdot n + \frac{145}{284} \cdot n\right) \cdot \log n \cdot (1 - \varepsilon).$$

If there is a vertex cover in G of size $\leq 144/284n$, then there is a tree with log-likelihood $\geq -(1.5 \cdot n + (144/284) \cdot n) \cdot \log n \cdot (1 + \varepsilon)$. Let

$$L_{d2}(\varepsilon) = -\left(m + \frac{145}{284} \cdot \frac{2 \cdot m}{3}\right) \cdot \log m \cdot (1 - \varepsilon), \quad \text{and}$$

$$L_{u2}(\varepsilon) = -\left(m + \frac{144}{284} \cdot \frac{2 \cdot m}{3}\right) \cdot \log m \cdot (1 + \varepsilon).$$

CLAIM 3.9. *Since in the reduction in Theorem 3.8, $2m/3 < n < 2m$, we can use the same reduction to show that $GL - ML[L_{d2}(\varepsilon), L_{u2}(\varepsilon)]$ is NP-hard.*

3.3 Approximation for sparse inputs

In this section, we consider a special type of inputs, termed sparse, under the Jukes–Cantor substitution model (Jukes and Cantor, 1969; Tuffely and Steel, 1997). We give an algorithm that finds a tree structure which approximates maximum log-likelihood for this model. Assuming an efficient local ML algorithm for optimizing the edge length of a given tree, the resulting tree approximates maximum log-likelihood for sparse inputs. The high level of the approximation is as follows: We first run the approximate parsimony algorithm, producing an ‘initial tree’. Now, we identify one of the edges going to one of the leaves as a root. While the tree has a large subforest (large will be defined soon), we uproot such a subforest and connect all its components directly to the root. When we terminate, the modified tree has only small

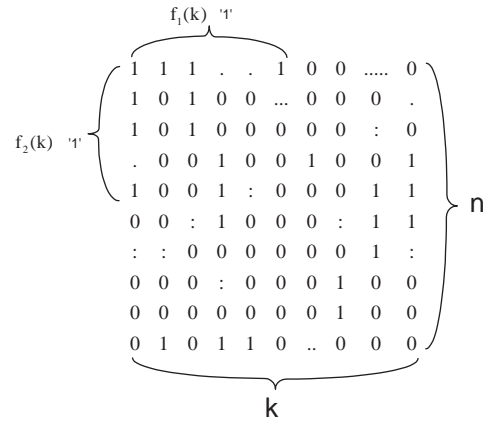


Fig. 2. Sparse input the for ML problem.

subforests hanging off its root. We now run the local ML algorithm on this tree. The resulting weighted tree is the output, and we claim its log-likelihood approximate $\log ML$.

DEFINITION 3.10. *Let n, k, c denote the number of input strings, the length of these strings, and the number of possible characters states, respectively. An input for the ML problem is sparse, if the following properties hold (see Fig. 2, where $x = 0$): There are two slow functions of $k, f_1, f_2 : N \mapsto N$, and a character state, x , such that every input string has a few characters that differ from x , and every site has a few characters that differ from x . More precisely*

1. each string contains no more than $f_1(k)$ sites whose state is different from x ;
2. in each site there are no more than $f_2(k)$ strings whose state at the site is different from x ;
3. the functions f_1, f_2 satisfy $f_2^5(k) \cdot f_1^2(k) = o(\log(k))$.¹

We note that the strings generated by our reduction from vertex cover are sparse with respect to the functions $f_1(k) = 2, f_2(k) = 3$. Consequently

OBSERVATION 3.11. *ML for sparse inputs is NP-hard.*

LEMMA 3.12. *Given a sparse input for the ML problem under the Jukes–Cantor substitution model, we can translate it to an equivalent sparse input, where one of the input strings is the all zero string.*

Equivalence means the same likelihood. We omit the (easy) proof.

LEMMA 3.13. *The parsimony score for a tree with n different strings at its leaves is at least $n - 1$.*

PROOF. Omitted.

¹The third condition can be relaxed: for every $\varepsilon > 0, f_2^{4+\varepsilon}(k) \cdot f_1^2(k) = o(\log(k))$. We chose $f_2^5(k)$ to simplify the proof.

In this section (unlike Section 3.1) the root is defined at distance zero from the all zero leaf. The following lemma will be used for proving the approximation ratio of our algorithm. The lemma suggests that for sparse inputs, there is a tree whose log-likelihood is arbitrarily close to the log-likelihood of the ML tree, and it can be rooted such that all the leaves in the tree are relatively close to the root. To achieve this, we use a process similar to the proof in Chor and Tuller (2005). Given an ML tree, we uproot its subforests, and connect them to the root. The main difference is the size of these subforests. In Chor and Tuller (2005) their size is $\log \log(k)$, while here we will take them to be of size $f_2^5(k)$.

LEMMA 3.14. *Let $S = [s(1), s(2), \dots, s(n)]$ be a sparse input to the ML problem, where S_1 is the all zero vector. Suppose each site contains less than $f_2(k)$ non-zero characters, and each string contains less than $f_1(k)$ non-zero characters. Then there is a tree such that the size of each subtree rooted at its root is at most $f_2^5(k)$, and for k large enough the ratio of its log-likelihood and the log-likelihood of the ML tree is arbitrarily close to 1.*

PROOF. We start with a general ML tree, and look for a subforest whose size is in the range $[f_2^5(k)/2, f_2^5(k)]$. When such a subforest is found, we uproot it and move it to the root. We modify Theorem 5.7 in Chor and Tuller (2005), by allowing $f_2^5(k)$ non-zero characters per site, instead of 3. We can now show that in each such step, the degradation in the parsimony score is at most $2(f_2(k) + 1)^4$. By a direct calculation, every such subtree has at most $f_2^5(k) \cdot f_1(k)$ non-zero characters. By Tuffely and Steel (1997), this implies that in each step the log-likelihood is decreased by at most

$$\Delta = 2 \log(k)(f_2(k) + 1)^4 + (f_2^5(k) \cdot f_1(k))^2.$$

The total log-likelihood decrease is at most $L_\Delta \leq 2n\Delta / f_2^5(k)$, namely

$$L_\Delta \leq \frac{2n(2 \log(k)(f_2(k) + 1)^4 + (f_2^5(k) \cdot f_1(k))^2)}{f_2^5(k)}.$$

Out of the k sites per string in the $f_2^5(k)$ leaves, at most $f_1(k)f_2^5(k)$ sites are non-zero. Thus for large enough k , the MP tree for each subforest is its ML tree: by Lemma 3.13, the total parsimony score of the final tree is $\Omega(n)$, thus the log-likelihood of this tree, and of the ML tree is $O(-n \cdot \log(k))$. The proof follows since by the sparsity of the input, $L_\Delta / (n \cdot \log(k)) \rightarrow 0$.

COROLLARY 3.15. *For sparse inputs, a MP tree under the constraint that each subtree, rooted at the root, has less than $f_2^5(k)$ leaves, can be augmented with edge lengths to give a $1 + o(1)$ approximation to the log-likelihood of input sequences.*

We propose the following algorithm, MLstruct, for finding an ML tree structure, given an algorithm for approximating MP:

1. Find the topology of the approximate MP tree.
2. While the root has subtrees with more than $f_2^5(k)$ leaves, uproot subforests of size $f_2^5(k)$ and move them to the root.
3. Use hill climbing or other local heuristic search for finding the length $q(e)$ of the edges in the resulting tree.

We can guarantee the algorithm will find a structure, such that if the edge lengths are optimal the approximation ratio is constant. We do not deal here with edge length optimization, where the standard approach is local heuristic search.

The approximation ratio of the algorithm stems from Corollary 3.15, the fact that log-likelihood ratio between the most parsimonious tree and the tree our algorithm finds in stage 2, is arbitrarily close to 1 for large enough k and the approximation ratio of the parsimony algorithm itself:

LEMMA 3.16. *Suppose the parsimony algorithm has approximation ratio A , and the local ML hill climbing algorithm is optimal. For $k \rightarrow \infty$, MLstruct achieves an approximation ratio*

$$\frac{\log(\Pr(S|T^{\text{MLstruct}}))}{\log(\Pr(S|T^{\text{ML}}))} \rightarrow_{k \rightarrow \infty} A.$$

PROOF. By Lemma 3.14 there is a tree such that the size of each subtree rooted with the root is less than $O((2 \cdot f_2(k) + 1)^4)$ and the ratio between the log-likelihood of the tree and the ML tree is $o((n \cdot \log(k) + L_\Delta) / (n \cdot \log(k)))$.

This is since we end up with a tree with log-likelihood greater than $\Omega(n \cdot \log(k))$, and the total decrease in the log-likelihood is at most L_Δ . According to Theorem 4.5 in Chor and Tuller (2005) and by Tuffely and Steel (1997) the log-likelihood of the tree, T_c^{ML} , with the best likelihood under the constraint that the size of each subtree rooted at the root is $O(f_2^4(k))$, is: $\text{Pars}(T_c^{\text{ML}}) \cdot \log(k) + o(\text{Pars}(T_c^{\text{ML}}) \cdot \log(k)) = \Omega(n \cdot \log(k))$.

The ratio between the parsimony of the truncated tree found by the approximating algorithm and the tree with the best parsimony under the constraint that the size of each subtree rooted with the root is, at most, $O((2 \cdot f_2(k) + 1)^4)$, and is bounded from above by:

$$n \cdot \left(1 + \frac{2 \cdot (f_2(k) + 1)^4}{(f_2(k))^5}\right) / n \rightarrow 1.$$

Consequently, if we start with a tree whose parsimony score is $A \cdot \text{OPT}$, and k is large enough, we get approximation ratio arbitrarily close to A for the log-likelihood.

COROLLARY 3.17. *By using, for example, the 2-approximation algorithm of Bonet et al. (1998), we get an*

approximation algorithm for log ML for sparse inputs with asymptotic approximating ratio of 2.

Let $C_p(n, k)$ be the complexity of the algorithm for finding the approximating MP tree. The complexity of finding the structure of the approximate ML tree is

$$C_p(n, k) + O\left(f_2^5(k) \cdot \left(n/f_2^5(k)\right)\right) = C_p(n, k) + O(n),$$

since we can find a subforest of size $O(f_2^5(k))$ for uprooting and moving to the root in $O(f_2^5(k))$ time. The complexity of finding edge length by hill climbing is discussed, for example, in Hendy *et al.* (1994).

The suggested algorithm is only a general framework for the actual algorithm. In practice, we apply the basic algorithm for each of the leaves as the root (the all-zero sequence), and try different sizes of the uprooted subforests, in order to increase the likelihood of the tree compared to the initial tree (found by the MP algorithm). We choose the final tree with the best likelihood.

The overall complexity of the algorithm remains polynomial even when applying these steps.

4 SIMULATED AND REAL DATASETS

4.1 Synthetic datasets

In order to evaluate our method we first used the following procedure for generating synthetic datasets:

We chose a phylogenetic tree topology of five taxa (see Fig. 3), and the two states model of Neyman. We repeated the following step five times. Each time we changed the constant \max_e , which affects the number of constant positions in the sampled dataset (we checked \max_e in the range 0.1..0.5).

Repeat the following steps 100 times:

- Sample the length of each edge in the tree from the uniform distribution $[0, \max_e]$.
- Sample sequences of length 10 000 sites.
- Construct a solution tree by our algorithm.
- Compare the likelihood of the constructed tree to that of the ML tree, by checking all the possible tree structures.

The results are summarized in Table 1. For each dataset (a different \max_e , that is, different average number of constant sites) we calculated: the failure probability of our method to find the ML tree, the average ratio between the log-likelihood of the tree of our method and the ML tree, and the worse ratio between the log-likelihood of the tree our method find and the ML tree. Our method had the best performances when the percentage of constant sites is high (above 70%, corresponding to $\max_e = 0.1$), but became worse for lower percentage of constant sites.

In the second test, we ran our method on synthetic datasets of Hendy and Holland (2003). This dataset was constructed by sampling trees which satisfy the molecular clock assumption.

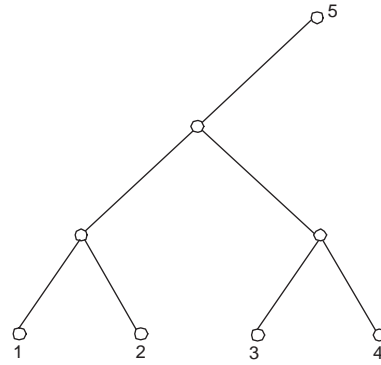


Fig. 3. The tree used for generating Monte Carlo samples.

Table 1. The results of implementing our method on synthetic datasets

% Const sites	73	53	39	32	25
aver approx	1.00001	1.0004	1.0008	1.00005	1.0001
worse approx	1.0005	1.0389	1.0758	1.0027	1.0027
Prob \neq ML	0.02	0.03	0.06	0.04	0.04

We generated random datasets each with different ratio of constant sizes, the average approximation ratio, worse approximations ratio, and calculated the probability of not finding the ML structure for each dataset. Our method has the best performances when the percentage of constant sites is high (above 70% or $\max_e = 0.1$), but become worse for lower rate of constants sites.

Table 2. The results of implementing our method on synthetic datasets of (Hendy and Holland, 2003)

Topology	Average approx	Worse approx	Prob \neq ML
Comb	1.001102	1.025427	0.478906
Fork	1.000170	1.018377	0.082715
Giraffe	1.000420	1.025712	0.226758

The dataset includes samples from different tree topologies (Comb, Fork and Giraffe). For each topology we checked 256 different sets of parameters which are described in table 1 of Hendy and Holland (2003). For each set of parameters we sampled 10 inputs of lengths 100, 200, 400 and 800. We emphasise that some of the inputs here were not sparse. For each of the inputs we used our method, we compared the likelihood of the constructed tree our method found to that of the ML tree, by checking all possible tree structures. The results are summarized in Table 2. For each dataset (which was sampled from different tree topology) we calculated the probability that our method does not construct the ML tree, the average ratio between the log-likelihood of the tree our of method and the ML tree, and the worse ratio between the log-likelihood of the tree our method find and the ML tree. Our method find trees—whose likelihood is very close to the likelihood of the ML—tree, the average approximation ratio

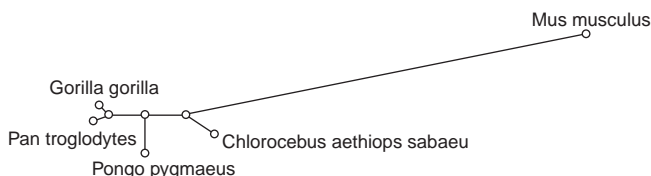


Fig. 4. The result of applying our algorithm for approximating the log-likelihood. The input to the algorithm is the α fucosyltransferase gene of five mammals.

is 1.001102, 1.000170 and 1.000420, for the Comb, Fork, and Giraffe topologies respectively.

The next stage was comparing our method to PHYLIP: We generated a few sparse datasets of 20 sequences and compared the performances of our method to the performances of PHYLIP. For these datasets we got similar performance, namely the two methods found trees with similar log-likelihood.

4.2 Biological datasets

In this section we demonstrate the performances of our algorithm on a biological dataset. The dataset includes the α fucosyltransferase DNA sequences (length around 1400 bp). It was analyzed for the five species: Pan troglodytes (chimpanzee, GenBank/EBI accession numbers AB015634), Gorilla gorilla (gorilla, GenBank/EBI accession numbers AB015635), Pongo pygmaeus (orangutan, GenBank/EBI accession numbers AB015636), Chlorocebus aethiops sabaueus (vervet monkey, GenBank/EBI accession numbers D87934) and Mus musculus (mouse, GenBank/EBI accession numbers AF064792) (Apoil *et al.*, 2000). We used Jukes–Cantor model for DNA substitution. We applied the package MEGA (Kumar *et al.*, 1994) for finding the MP tree structure (or approximation of it). The algorithm of Fitch (1971) and the connection between the the log-likelihood of a tree and the parsimony score of the tree due to Tuffely and Steel (1997) were used for finding bounds on the quality of the result (when compared to the log-likelihood of the optimal tree). Given a tree structure, the lengths of the edges (q_e) were determined by hill climbing (as in other methods). The resulting tree is shown in Figure 4, the edges' lengths in the figure are scaled according to the edges length we found, and the log-likelihood of the tree is -2655.705947 .

In order to evaluate our result, we checked all the 15 possible topologies for these datasets; for each topology we searched the edge length spectrum for ML tree. We discovered that our tree is the ML tree (of course, the algorithm itself does not perform an exhaustive search).

5 DISCUSSION AND CONCLUSION

In this work, we further explored the computational and algorithmic aspects of ML reconstruction of phylogenetic

trees. We showed ML is NP-hard even when restricted to trees under *molecular clock*. We showed that there is no efficient approximation algorithm for log-likelihood that achieves approximation ratio lower than 1.00175 (unless $P \neq NP$). We developed an ML approximating algorithm for sparse inputs under the Jukes–Cantor model. The algorithm achieves approximation ratio A for the log-likelihood, given an algorithm for edge length optimization, and another algorithm with approximation ratio A for parsimony. The algorithm is of practical interest, since many biological datasets are sparse. Furthermore, by combining approximation algorithms and heuristics, one may gain practical performance that improves upon both.

It is interesting to compare our results, which establish the computational hardness of ML reconstruction, to a number of works in the PAC learning (Kearns *et al.*, 1994) and the phylogenetic algorithms communities (Erdos *et al.*, 1999a,b; Csuros and Kao, 2001). These works essentially show that if data is sampled according to a ‘well behaved’ tree (edges are neither too long nor too short), then with high probability the original tree can be efficiently reconstructed (up to a small error in edges' lengths), given only a short sample size. Where, then, does the intractability of ML reconstruction stem from? If the data is sampled from a model that substantially deviates from a tree, then these reconstruction methods do not guarantee anything. Indeed, the ‘data’, or sequences, in our reduction are *not* induced by a tree. But what happens if the data *is* induced by a tree? It is well known that ML is *consistent*, meaning that if the sample size is large enough, the ML tree will be the original tree (Chang, 1996). But how large is ‘large enough’? The best upper bound to date, by Steel and Szekely (2002), is exponential in the number of species. If this bound is indeed tight, then ML reconstruction may be hard even if the data is induced by a tree. (Preliminary results that the likelihood function has multiple maxima even for tree-generated data may support such conjecture.) If, however, the truth is that a polynomially long sample size suffices to guarantee that the ML tree will be the original tree with high probability, then the above mentioned methods do in fact find the ML tree for such well behaved data.

Interesting questions remain open. What about approximation algorithms of log-likelihood for *general* inputs? How does our method perform for non-sparse inputs? It is desirable to identify regions of the sequence space where ML is *tractable*. Here we ignored the complexity of finding edge length for a given tree structure. But in actuality, it is not even known what the complexity of the problem is 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. Multiple maxima (Steel, 1994; Chor *et al.*, 2000) shed doubts on the (worst case) correctness of this approach.

ACKNOWLEDGEMENTS

We wish to thank Tal Pupko, Metsada Pasmanik-Chor, and Mike Steel for helpful discussions. This research was supported by ISF grant 418/00.

REFERENCES

- Addario-Berry,L., Chor,B., Hallett,M., Lagergren,J., Panconesi,A. and Wareham,T. (2004) Ancestral maximum likelihood of evolutionary trees is hard. *J. Bioinform. Comput. Biol.*, **2**(3), 257–271.
- Apoil,P., Roubinet,F., Despiau,S., Mollicone,R., Oriol,R. and Blancher,A. (2000) Evolution of 2-fucosyltransferase genes in primates: relation between an intronic Alu-Y element and red cell expression of ABH. *Mol. Biol. Evol.*, **17**, 337–351.
- Ausiello,G., Crescenzi,P., Gambosi,G., Kann,V., Marchetti-Spaccamela,A. and Protasi,M. (1998) *Complexity and Approximation*, Springer, Berlin.
- Berman,P. and Karpinski,M. (1999) On some tighter inapproximability results. In *Proceedings of the 26th ICALP (ICALP 1999)*, pp. 200–209.
- Bonet,M., Steel,M., Warnow,T. and Yooshef,S. (1998) Faster algorithms for solving parsimony and compatibility. *J. Comput. Biol.*, **5**(3), 409–422.
- Chang,J.T. (1996) Full reconstruction of Markov models on evolutionary trees: identifiability and consistency. *Math. Biosci.*, **137**, 51–73.
- Chor,B., Hendy,M.D., Holland,B.R. and Penny,D. (2000) Multiple maxima of likelihood in phylogenetic trees: an analytic approach. *Mol. Biol. Evol.*, **17**, 1529–1541.
- Chor,B. and Tuller,T. (2005) Maximum likelihood of evolutionary trees is hard. In *Proceedings of the 9th Annual International Conference on Research in Computational Molecular Biology (RECOMB05)*, May 2005.
- Csuros,M. and Kao,M.Y. (2001) Provably fast and accurate recovery of evolutionary trees through harmonic greedy triplets. *SIAM J. Comput.*, **31**, 306–322.
- Day,W. (1987) The computational complexity of inferring phylogenies from dissimilarity matrix. *Bull. Math. Biol.*, **49**, 461–467.
- Day,W. and Sankoff,D. (1986) The computational complexity of inferring phylogenies by compatibility. *Syst. Zool.*, **35**, 224–229.
- Day,W., Johnson,D. and Sankoff,D. (1986) The computational complexity of inferring rooted phylogenies by parsimony. *Math. Biosci.*, **81**, 33–42.
- Erdos,P.L., Steel,M.A., Szekely,L.A. and Warnow,T.J. (1999a) A few logs suffice to build (almost) all trees: Part 1. *Random Struct. Algorithms*, **221**, 153–184.
- Erdos,P.L., Steel,M.A., Szekely,L.A. and Warnow,T.J. (1999b) A few logs suffice to build (almost) all trees: Part 2. *Theoret. Comput. Sci.*, **221**, 77–118.
- Felsenstein,J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, **17**, 368–376.
- Felsenstein,J. (1996) Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. In Doolittle,R.F. (ed.), *Computer Methods for Macromolecular Sequence Analysis Methods in Enzymology*, Academic Press, Orlando, Florida, Vol. 266, pp. 418–427.
- Fitch,W.M. (1971) Toward defining the course of evolution: minimum change for specified tree topology. *Syst. Zool.*, **20**, 406–416.
- Foulds,L. and Graham,R. (1982) The steiner problem in phylogeny is NP-complete. *Adv. Appl. Math.*, **3**, 43–49.
- Gaunt,M.W. and Miles,M.A. (2002) An insect molecular clock dates the origin of the insects and accords with palaeontological and biogeographic landmarks. *Mol. Biol. Evol.*, **19**, 748–761.
- Goodman,M. (1981) Globin evolution was apparently very rapid in early vertebrates: a reasonable case against the rate-constancy hypothesis. *J. Mol. Evol.*, **17**, 114–120.
- Graur,D. and Li,W.H. (1999) *Fundamentals of Molecular Evolution*, 2nd edn. Sinauer, Sunderland, MA.
- Hendy,M.D. and Holland,B.R. (2003) Upper bounds on maximum likelihood for phylogenetic trees. *Bioinformatics*, **19**, 66–72.
- Hendy,M.D. and Penny,D. (1993) Spectral analysis of phylogenetic data. *J. Classif.*, **10**, 5–24.
- Hendy,M.D., Penny,D. and Steel,M.A. (1994) A discrete fourier analysis for evolutionary trees. *Proc. Natl Acad. Sci. USA*, **91**, 3339–3343.
- Jukes,T.H. and Cantor,C.R. (1969) Evolution of protein molecules. In Munro,H.N. (ed.), *Mammalian Protein Metabolism*. Academic Press, NY, pp. 21–132.
- Karpinski,M. (2001) Approximating bounded degree instances of NP-hard problems. In *Proceedings of the 13th Symposium on Fundamentals of Computation Theory (FCT), LNCS 2138*. Springer, Berlin.
- Kearns,M., Mansour,Y., Ron,D., Rubinfeld,R., Schapire,R.E. and Sellie,L. (1994) On the learnability of discrete distributions. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC94)*. ACM, New York.
- Kumar,S., Tamura,K. and Nei,M. (1994) MEGA: Molecular Evolutionary Genetics Analysis software for microcomputers. *Comput. Appl. Biosci.*, **10**, 189–191.
- Miyamoto,M., Koop,B.F., Slightom,J.L., Goodman,M. and Tennant,M.R. (1988) Molecular systematics of higher primates: Genealogical relations and classification. *Proc. Natl Acad. Sci. USA*, **85**, 7625–7631.
- Margoliash,E. (1963) Primary structure and evolution of cytochrome. *Proc. Natl Acad. Sci. USA*, **50**, 672–679.
- Neyman,J. (1971) Molecular studies of evolution: a source of novel statistical problems. In Gupta,S. and Jackel,Y. (ed.), *Statistical Decision Theory and Related Topics*. Academic Press, NY, pp. 1–27.
- Nei,M., Kumar,S. and Kumar,S. (2000) *Molecular Evolution and Phylogenetics*. Oxford University Press, NY.
- Papadimitriou,C.H. (1993) *Computational Complexity*. Addison-Wesley, Redwood city, CA.
- Steel,M. (1992) The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classif.*, **9**, 71–90.
- Steel,M. (1994) The maximum likelihood point for phylogenetic tree is not unique. *Syst. Biol.*, **43**, 560–564.
- Steel,M. and Penny,D. (2000) Parsimony, likelihood and the role of models in molecular phylogenetics. *Mol. Biol. Evol.*, **17**, 839–850.
- Steel,M. and Szekely,L.A. (2002) Inverting random functions (2): explicit bounds for discrete maximum likelihood estimation, with applications. *SIAM J. Discr. Math.*, **15**, 562–575.
- Tuffely,C. and Steel,M. (1997) Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bull. Math. Biol.*, **59**, 581–607.

- Wareham,T. (1993) On the computational complexity of inferring evolutionary trees. *Technical Report 93-01*, Department of computer science, Memorial University of Newfoundland.
- Yoder,A.D. Yang,Z. (2000) Estimation of primate speciation dates using local molecular clocks. *Mol. Biol. Evol.*, **17**, 1081–1090.
- Zuckerandl,E. and Pauling,L. (1962) Molecular disease, evolution and genetic heterogeneity. In Kasha,M. and Pullman (eds), *Horizons in Biochemistry*, Academic Press, NY, pp. 189–225.
- Zuckerandl,E. and Pauling,L. (1965) Evolutionary divergence and convergence in proteins. In Bryson,V. and Vogels,H.J. (eds), *Evolving Genes and Proteins*. Academic Press, NY, pp. 97–166.