PHYLOGENETIC ANALYSIS VIA DC PROGRAMMING

STEVEN E. ELLIS* AND MADHU V. NAYAKKANKUPPAM†

Abstract. The evolutionary history of species may be described by a *phylogenetic tree* whose topology captures ancestral relationships among the species, and whose branch lengths denote evolution times. For a fixed topology and an assumed probabilistic model of nucleotide substitution, we show that the likelihood of a given tree is a d.c. (difference of convex) function of the branch lengths, hence maximum likelihood estimates (of the branch lengths) may be obtained by solving an appropriate d.c. program. Such a formulation is amenable to global optimization techniques, in contrast with existing methods and software codes which potentially produce only locally optimal solutions. We present the formulation of this optimization problem, its solution via an outer approximation cutting plane algorithm, and illustrative numerical results on small genetic data sets.

Key words. Phylogenetic analysis, maximum likelihood estimation, d.c. programming, cutting plane algorithm, global optimization.

AMS subject classifications. 92D20, 90C26, 90C90.

1. Introduction.

1.1. Background and Terminology. A fundamental task in evolutionary biology is the inference of phylogenetic or ancestral relationships among contemporary species. Every organism's genetic blueprint is encoded by the genes in its DNA (a sequence of nucleotides, each being a symbol in $\mathcal{N} = \{A, C, T, G\}$), and its biological functions are carried out by means of proteins (a sequence of amino acids, each being one of 20 possible symbols). Owing to nucleotide substitutions in DNA, altered genes and proteins are produced over a period of time, and a hierarchy of organisms with somewhat different DNA and proteins evolves. This process of biological evolution may be encoded by a *phylogenetic tree* — a bifurcating tree with leaf nodes (denoting observed, contemporaneous species) deriving from internal nodes (denoting ancestral species). The topology of this tree encodes the evolutionary relationships among the observed species, while the length of a branch between two adjacent nodes in the tree is a measure of the elapsed time (*divergence time*) for the species at one node to evolve into the species at the other node. The goal of *phylogenetic analysis* is the reconstruction of this phylogenetic tree, both its topology and its branch lengths, from the observed DNA sequences of the contemporary species at the leaf nodes.

Three widely-used methods for phylogenetic analysis are (i) parsimony, (ii) distance-based methods and (iii) maximum likelihood estimation. Experimental simulations [14] indicate that maximum likelihood (ML) estimates of phylogenetic trees are consistently superior to parsimony or distance-based methods. The distinct advantage of likelihood-based methods lies in their ability to provide quantitative and statistically meaningful measures of confidence in the reconstruction. We focus on this method in the remainder of the paper. In the ML approach, we start with an underlying probabilistic model of nucleotide substitution, usually a Markov chain model derived from instantaneous rates of transition of one nucleotide symbol into another. Then,

*Department of Mathematics & Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore MD 21250. Email: ells1@umbc.edu. Supported in part by a REU award under NSF grant DMS-0238008 and a UMBC Provost's Undergraduate Research Award.

†Department of Mathematics & Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore MD 21250. Email: madhu@math.umbc.edu. Supported in part by NSF grants DMS-0238008, DMS-0215373 and a grant from the UMBC Designated Research Initiative Fund.

the ML methodology infers the most likely topology and branch lengths which are consistent with the given DNA sequence data and the assumed probabilistic model of nucleotide substitution. This is achieved by maximizing a *likelihood function* $L(x;\tau)$ jointly over the vector of branch lengths x (continuous variables specifying the length of each edge in the tree) and the topology τ (combinatorial variable). This likelihood maximization is difficult even if one of the two variables, x or τ, is held fixed:

- **Branch Length Optimization:** Even for a fixed topology τ, and for most reasonable probabilistic models of nucleotide substitution, the likelihood function $L(x;\tau)$ is a highly nonlinear function of the branch lengths x. The heuristic approaches currently in use may get trapped in local maxima.
- **Topology Optimization:** Even for a fixed choice of the branch lengths x, the number of topologies grows factorially in the number of input sequences, making exhaustive search futile. It is known [1] that the number of such topologies $T(n)$ on n input sequenes is given by $T(n) = (2n-5)!/[(n-3)!2^{n-3}]$. (With 53 input sequences, this number exceeds the total number of atoms in the visible universe, estimated to be over 4×10^{78}.)

Thus the problem of computing globally optimal ML estimates of phylogenetic trees is a computationally intractable one; see the recent work [3]. Existing methods[1] in the literature work around this in different ways. Some algorithms resort to stochastic techniques (*e.g.* simulated annealing, genetic algorithms, Markov Chain Monte Carlo), while others rely more on deterministic optimization. PHYLIP [8] uses the *stepwise addition* heuristic (explained in Section 6) to generate the topology incrementally, while using a coordinate optimization method to optimize branch lengths. PAML [25] provides several options for topology generation (including stepwise addition, or simply fixing a user-supplied topology), and uses a quasi Newton method or the conjugate gradient algorithm with a Wolfe line search to optimize branch lengths. However, even for a fixed topology, these methods are not guaranteed to produce globally optimal estimates of branch lengths.

1.2. Summary. In this paper we primarily focus on computing globally optimal ML estimates of branch lengths for a given topology using d.c. programming. To this end, we first list well known properties of d.c. functions and give a simple outer-approximation algorithm for d.c. programming in Section 2. These properties are then used to derive a d.c. formulation of the (log) likelihood function (Section 3), which may then be globally maximized by the given algorithm; related implementation details are discussed in Section 4. Numerical calculations (Section 5) on small problem instances and comparisons with existing codes confirm the validity of the approach. In Section 6, we briefly indicate how the proposed approach may be incorporated into existing, practically successsful heuristic procedures for determining good estimates of the optimal topology.

2. DC Programming. We summarize a few basic facts about d.c. functions, then describe a simple outer-approximation based cutting plane method for optimizing them.

2.1. DC functions. A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be d.c. on X if

$$f(x) = g(x) - h(x) \; \forall x \in X, \tag{2.1}$$

[1] J. Felsenstein's PHYLIP [8] web page is an excellent resource with links [6] to over 200 available packages for phylogenetic inference.

with $g, h : X \to \mathbb{R}$ being convex functions on X. Such a decomposition, when it exists, is clearly nonunique. Nevertheless, even without an explicit decomposition at hand, a large class of functions may be readily identified as d.c. via Hartman's deep result [10] that every locally d.c. function (*i.e.* one that has a possibly different d.c. decomposition on an open neighborhood of each point in $\mathbb{R}^n$) is also globally d.c. (*i.e.* admits a d.c. decomposition globally valid on $\mathbb{R}^n$). Thus every $\mathcal{C}^2$ function f must be d.c., since for suitably large $K > 0$,

$$f(x) = \left[f(x) + K\|x\|^2\right] - K\|x\|^2$$

is locally (hence globally) d.c. Consequently, polynomials are d.c., and we may therefore conclude that continuous functions may be approximated (uniformly on compact subsets) by d.c. functions.

Yet algorithms for optimizing d.c. functions rely on an explicit d.c. decomposition being available. We list several well known operations on d.c. functions that preserve the d.c. structure with explicitly computable d.c. decompositions; these and other related results may be found in, for instance, [12, 17, 24]. Some of these results will be used in Section 3.

PROPOSITION 2.1 (Properties of d.c. functions).

1. *If $f = g - h$ and $f_i = g_i - h_i$ $(i = 1, , ..., m)$ are d.c. functions, then so also are:*

$$\sum_{i=1}^{m} \lambda_i f_i = \left[\sum_{\{i:\lambda_i \geq 0\}} \lambda_i g_i - \sum_{\{i:\lambda_i < 0\}} \lambda_i h_i\right] - \left[\sum_{\{i:\lambda_i \geq 0\}} \lambda_i h_i - \sum_{\{i:\lambda_i < 0\}} \lambda_i g_i\right]$$

$$\max_{1\leq i\leq m} f_i = \max_{1\leq i\leq m} \left\{g_i + \sum_{j=1,\, j\neq i}^{m} h_j\right\} - \sum_{j=1}^{m} h_j$$

$$\min_{1\leq i\leq m} f_i = \sum_{j=1}^{m} g_j - \max_{1\leq i\leq m} \left\{h_i + \sum_{j=1,\, j\neq i} g_j\right\}$$

$$|f| = 2\max\{g, h\} - \{g + h\}$$

$$f^+ = \max\{0, f\}$$

$$f^- = \min\{0, f\}$$

2. *Let f_1 and f_2 be nonnegative d.c. functions. Then the product $f_1 \cdot f_2$ is d.c. with the following d.c. decomposition:*

$$f_1 \cdot f_2 = \frac{1}{2}\left[(g_1 + g_2)^2 + (h_1 + h_2)^2\right] - \frac{1}{2}\left[(g_1 + h_2)^2 + (g_2 + h_1)^2\right].$$

3. *Let $f(x)$ be a d.c. function defined on a compact convex set $X \subset \mathbb{R}^m$ such that $f(x) \geq a > 0$ $\forall x \in X$. If $q : [a, \infty] \to \mathbb{R}$ is a convex nonincreasing function such that $q'_+(a) > -\infty$, then $q(f(x))$ is a d.c. function on X:*

$$q(f(x)) = p(x) - K\left[g(x) + h(x)\right]$$

where $p(x) = q(f(x)) + K\left[g(x) + h(x)\right]$ is a convex function and K is a constant satisfying $K \geq \left|q'_+(a)\right|$.

2.2. DC optimization. Here we state an outer approximation cutting plane method from [13] to solve

$$\min_x \quad f(x) = g(x) - h(x) \quad \text{s.t. } x \in X = \{x \in \mathbb{R}^n : \alpha(x) \leq 0\}, \tag{2.2}$$

where $\alpha : \mathbb{R}^n \to \mathbb{R}$ is a convex function, and the convex feasible set X is assumed to be compact with nonempty interior. We assume that f has an explicit d.c. decomposition on X in terms of known convex functions g and h. The algorithm is based on the optimality condition that $x^* \in X$ is optimal for (2.2) if and only if there exists $t^* \in \mathbb{R}$ such that

$$0 = \inf \{-h(x) + t : x \in X,\ t \in \mathbb{R},\ g(x) - t \leq g(x^*) - t^*\}. \tag{2.3}$$

Starting with an initial guess $y^0 \in \operatorname{int} X$, Algorithm 2.2 generates a sequence y^k, of which any accumulation point x^* is an optimal solution of (2.2); see [13] for a convergence proof.

Methods to construct the first polytope P^0 and its vertex set $V(P^0)$ are given in [13]. A procedure to compute the vertex set of the reduced polytope P^{k+1} in Step 20 is given in [12]. This algorithm, chosen for its simplicity and ease of implementation, does not involve any computationally intensive subproblems — only function and subgradient evaluations, vertex enumeration, and whenever Step 16 is encountered, a root finding procedure for a univariate convex function. Other types of algorithms for d.c.programming are available in [12, 13, 22–24].

3. ML Estimation of Branch Lengths. Beginning with an underlying probabilistic model of nucleotide substitution, we derive a d.c. representation of the likelihood function to show how the problem of ML estimation of branch lengths may be formulated and solved as a d.c. program.

3.1. Nucleotide substitution model. A standard probabilistic model for nucleotide evolution is given by a Markov process with a specified instantaneous rate at which a nucleotide i gets substituted by nucleotide j. This rate, call it q_{ij}, is taken to be proportional to a mean rate μ (common to all nucleotides) as well as to π_j, the equilibrium frequency of nucleotide j:

$$q_{ij} \propto \mu\pi_j.$$

Introducing appropriate proportionality constants, these rates may be collectively specified by an *instantaneous substitution rate matrix*

$$Q = \begin{bmatrix} \cdot & \mu a\pi_C & \mu b\pi_G & \mu c\pi_T \\ \mu g\pi_A & \cdot & \mu d\pi_G & \mu e\pi_T \\ \mu h\pi_A & \mu j\pi_C & \cdot & \mu f\pi_T \\ \mu i\pi_A & \mu k\pi_C & \mu l\pi_G & \cdot \end{bmatrix}, \tag{3.1}$$

whose diagonal entries are inferred from the condition that rows must sum to zero. The matrix Q is the infinitesimal generator for the Markov chain modeling probabilistic nucleotide substitutions: the (i, j) entry of Q is the expected number of substitutions of nucleotide i by nucleotide j in an infinitesimal time interval dt.

In practice, the equilibrium frequencies are chosen equal ($\pi_A = \pi_C = \pi_T = \pi_G = 0.25$), or are taken to be the empirical frequencies observed in the data. We assume that the other parameters (μ and the proportionality constants $a, \ldots, l$) are given, hence Q is completely specified.

Algorithm 2.1 Outer Approximation Cutting Plane Method [13]

1: **Initialization:** Set $\omega^0 = g(y^0) - h(y^0)$, the first upper bound of the optimal value $\omega^* = f(x^*) = g(x^*) - h(x^*)$ of (2.2).
2: Compute a subgradient $s \in \partial g(y^0)$ to construct the affine function $l(x) = (x - y^0)^T s + g(y^0)$.
3: Construct a simplex $S^0 \supseteq X$ with vertex set $V(S^0)$. Choose $\bar{\omega}$ and $\bar{t}$ such that

$$\bar{\omega} = \min \left\{ l(x) : x \in V(S^0) \right\} - \max \left\{ h(x) : x \in V(S^0) \right\}$$
$$\bar{t} > \max \left\{ g(x) : x \in V(S^0) \right\}.$$

This ensures that the function $\beta^k(x,t) = \max \left\{ \alpha(x),\ g(x) - t - \omega^k \right\}$ satisfies $\beta^k(y^0, \bar{t}) < 0 \ \ \forall \omega^k \leq \omega^0$.
4: Construct a polytope $P^0 \supseteq \{(x,t) : x \in X,\ t \in \mathbb{R}^n,\ g(x) - t - \omega^* = 0\}$, and compute this initial polytope's vertex set $V(P^0)$.
5: Set $k = 0$.
6: **Iteration:** Compute an optimal solution (x^k, t^k) of the problem $\min \left\{ -h(x) + t : (x,t) \in V(P^k) \right\}$.
7: **if** $-h(x^k) + t^k = 0$ **then**
8: Stop: y^k is the optimal solution of(2.2) with optimal value ω^k.
9: **else**
10: **if** $x^k \in X$ (feasible case) **then**
11: Compute $s^k \in \partial g(x^k)$.
12: Compute the improved upper bound $\omega^{k+1} = \min \left\{ \omega^k, g(x^k) - h(x^k) \right\}$, taking y^{k+1} such that $g(y^{k+1}) - h(y^{k+1}) = \omega^{k+1}$.
13: **else**
14: Define the convex function $\beta^k(x,t) = \max \left\{ \alpha(x);\ g(x) - t - \omega^k \right\}$.
15: Compute $s^k \in \partial \beta^k(x^k, t^k)$.
16: We have $\beta^k(x^k, t^k) > 0$ and $\beta^k(y^0, \bar{t}) < 0$, so compute the zero (ζ^k, θ^k) of $\beta^k(x,t)$ on the line segment joint (x^k, t^k) and $(y^0, \bar{t})$.
17: Compute the improved upper bound $\omega^{k+1} = \min \left\{ \omega^k, g(\zeta^k) - h(\zeta^k) \right\}$, taking y^{k+1} such that $g(y^{k+1}) - h(y^{k+1}) = \omega^{k+1}$.
18: **end if**
19: Construct the cutting plane (affine function)

$$l^k(x,t) = \begin{cases} (x - x^k)^T s^k + g(x^k) - \omega^{k+1} - t, & \text{if } x^k \in X \\ \left((x,t) - (x^k, t^k)\right)^T s^k + \beta^k(x^k, t^k), & \text{if } x^k \notin X. \end{cases}$$

20: Set $P^{k+1} = P^k \cap \left\{ (x,t) : l^k(x,t) \leq 0 \right\}$, and compute $V(P^{k+1})$.
21: **end if**
22: Set $k = k + 1$, and iterate by returning to Step 6.

3.2. Problem setting and assumptions. We start with the given DNA sequences for n taxa, each with m sites, and a given unrooted, bifurcating tree topology τ. For $n \geq 3$ sequences, this means that τ has $2n-2$ nodes in all, with n leaf nodes (corresponding to the n taxa) of degree 1 and $n-2$ internal nodes of degree 3, and $2n-3$ interconnecting branches whose lengths are to be determined. We assume that these n sequences are multiply aligned, and that this $n \times m$ matrix of nucleotides is completely specified, each entry being one of A, C, T or G; see Table A.1 and Fig-

ure A.1 in Appendix A for an example. (We restrict our attention to DNA sequences, although the proposed approach — with minor modifications — is applicable to protein sequence data as well.) Each of these nucleotides evolves independently according to the same rate matrix (3.1), which is assumed to be time- and lineage-invariant, *i.e.* Q remains fixed in time and along different branches of the tree τ.

The problem of branch length estimation is to determine nonnegative values for the $2n-3$ branch lengths, collectively denoted by the vector x, that maximize the likelihood $L(x)$ (or equivalently, its logarithm) of the observed data at the leaf nodes of τ:

$$\max_{x \geq 0} \ \ln L(x). \tag{3.2}$$

3.3. DC decomposition of log likelihood. By imposing appropriate bound constraints on x, we show that the problem (3.2) of obtaining ML estimates of branch lengths may be formulated as a d.c. program:

$$\min_{0<l\leq x\leq u} \ -\ln L(x), \tag{3.3}$$

where the objective function $(-\ln L(x))$ has an explicit d.c. decomposition.

THEOREM 3.1 (Log likelihood is d.c.). *If Q is diagonalizable, then $-\ln L(x)$ is (explicitly) d.c. on $X := \{x \in \mathbb{R}^{2n-3} : 0 < l \leq x \leq u\}$.*

Proof. We proceed in four incremental stages culminating in an explicit d.c. decomposition for $-\ln L(x)$.

1. Substitution probabilities. Let $U = [u_1, \ldots, u_4]$ be a matrix whose columns are eigenvectors of Q corresponding to eigenvalues $\lambda_1, \ldots, \lambda_4$, and denote the rows of U^{-1} by $v_1^T, \ldots, v_4^T$. Then the matrix of substitution probabilities over a time interval t

$$P(t) = e^{Qt} = \sum_{k=1}^{4} e^{\lambda_k t} u_k v_k^T$$

is manifestly entrywise d.c.: its (i,j) entry

$$p_{ij}(t) = \sum_{k=1}^{4} e^{\lambda_k t} (u_k)_i \, (v_k)_j$$

is given by an explicitly weighted sum of exponentials.

2. Sitewise likelihood. Fix a site index s, so that the n leaf nodes of the given tree τ are labeled with nucleotides from the s^{th} site of the n input sequences. Let x be the vector of all branch lengths in τ. For any node $c \in \tau$, denote by $L_i^c(x;s)$ the conditional likelihood of node c having symbol i, given the data at the leaf nodes of the subtree rooted at node c. The following recursive argument uses Proposition 2.1 to show that this conditional likelihood is d.c. at each node of the tree. If c is a leaf node, then $L_i^c(x;s)$ is unity for that symbol i labeling this leaf node, and is zero for the three other symbols; this is trivially d.c. If c is an internal node connected to children nodes a and b via branches of lengths x_a and x_b respectively (see Figure 3.1), then

$$L_i^c(x;s) = L_i^c(x;s,a) \ \times \ L_i^c(x;s,b), \quad \text{where} \tag{3.4}$$

$$L_i^c(x;s,a) = \sum_{j \in \mathcal{N}} p_{ji}(x_a) L_j^a(x;s), \quad \text{and} \tag{3.5}$$

$$L_i^c(x;s,b) = \sum_{k \in \mathcal{N}} p_{ki}(x_b) L_k^b(x;s). \tag{3.6}$$

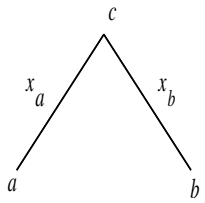

FIG. 3.1. *Typical tree segment.*

The product in (3.4) stems from the assumption that evolution occurs independently along branches. Both $L_i^c(x;s,a)$ and $L_i^c(x;s,b)$, as weighted sums of d.c. functions, are themselves d.c. Therefore $L_i^c(x;s)$, as a product of d.c. functions, must also be d.c. In particular, for the root node r of τ, $L_i^r(x;s)$ is d.c., hence also the likelihood of the entire tree for the s^{th} site

$$L^r(x;s) = \sum_{i\in\mathcal{N}} \pi_i L_i^r(x;s).$$

3. Sitewise log likelihood. Note that when all branch lengths x are strictly positive, so are the substitution probability functions p_{ij} and the likelihood $L^r(x;s)$ for site s. Again, by Proposition 2.1, the composite function $-\ln L^r(x;s)$ is d.c.
4. Log likelihood. Since sites are assumed to evolve independently, the log likelihood of the tree

$$-\ln L(x) = -\sum_{s=1}^{m} \ln L^r(x;s) \tag{3.7}$$

is also d.c. □

Several well known nucleotide substitution models result in a symmetric generator Q. Taking $a = \ldots = l = 1$ and $\pi_i = 0.25\ \forall i \in \mathcal{N}$ in Q yields the JC69 Jukes-Cantor model [15]:

$$p_{ij}(t) = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-\mu t} & (i=j) \\ \frac{1}{4} - \frac{1}{4}e^{-\mu t} & (i\neq j) \end{cases} \tag{3.8}$$

with substitution probability functions that are purely convex or concave. Kimura's two-parameter model K2P [16] divides the nucleotides into two groups — the purines (A,G) and the pyrimidines (C,T) — and allows for different relative rates of intra-group substitutions (known as *transitions*) and intergroup substitutions (known as *transversions*) by setting $a = c = d = f = g = j = i = l = 1$, but $b = e = h = k = \kappa$. With equal equilibrium base frequencies π_i, this model gives rise to

$$p_{ij}(t) = \begin{cases} \frac{1}{4} + \frac{1}{4}e^{-\mu t} + \frac{1}{2}e^{-\mu\left(\frac{\kappa+1}{2}\right)t}, & (i=j) \\ \frac{1}{4} + \frac{1}{4}e^{-\mu t} - \frac{1}{2}e^{-\mu\left(\frac{\kappa+1}{2}\right)t}, & (i\neq j,\ \text{transition}) \\ \frac{1}{4} + \frac{1}{4}e^{-\mu t}, & (i\neq j,\ \text{transversion}). \end{cases} \tag{3.9}$$

Both these models result in a symmetric Q, hence the Markov chain $P(t)$ modeling nucleotide substitutions is time-reversible. (One consequence of a time-reversible model is that the root node may be chosen arbitrarily.) Practically every model in the literature (including [5, 7, 11, 15, 16, 21, 26]) gives rise to a diagonalizable generator, hence this assumption in the theorem is not a serious limitation.

4. Computational Details. We discuss several computational issues pertaining to our MATLAB implementation. When solving (3.3) with a generic d.c. programming method such as Algorithm 2.2, using closed form expressions for the objective function and its gradient (via Theorem 3.1 and Proposition 2.1) is both inefficient and complicated; see (A.1) and (A.2) in Appendix A. Instead, computations of function and gradient values may be ordered to mimic the topology of the phylogenetic tree itself, *i.e.* given a point x, $-\ln L(x)$ in (3.7) is computed recursively with a post-order traversal of the tree, using (3.4) to compute intermediate values at each node. Simultaneously, gradient values are readily available via the chain rule:

$$\nabla_{x_a} L_i^c(x;s) = \left(\sum_{j \in \mathcal{N}} p'_{ji}(x_a) L_j^a(x;s) \right) \left(\sum_{k \in \mathcal{N}} p_{ki}(x_b) L_k^b(x;s) \right),$$

where we have used the fact that $L_j^a(x;s)$ (the conditional likelihood of the subtree rooted at node a) is independent of x_a (the branch length connecting node a to node c).

Since the algorithm relies on an explicit d.c. decomposition of the final log likelihood objective function and its gradient, their constituent intermediate values must also be represented as a difference of two quantities in a manner that is consistent with their implicit d.c. decompositions. For example, the probability of an intragroup transition in the K2P model (3.9) must be explicitly retained as an ordered pair

$$\left(\frac{1}{4} + \frac{1}{4} e^{-\mu t}, \;\; \frac{1}{2} e^{-\mu \left(\frac{\kappa+1}{2} \right) t} \right)$$

of quantities whose difference equals $p_{ij}(t)$. Results of subsequent computations involving this $p_{ij}(t)$, such as the conditional likelihoods $L_i^c(x;s)$ on each tree node, must also be stored as ordered pairs so that the final objects of interest, namely objective function and gradient values, are available in decomposed form.

Significant savings may be realized by vectorizing these computations. For instance, to evaluate $L_i^c(x;s,a)$ in (3.5) with the JC69 model (3.8), denote by $L^a(x;s) = u - v$, the d.c. decomposition of the 4-dimensional vector $L^a(x;s)$ whose i^{th} component is $L_i^a(x;s)$. Similarly decompose the JC69 substitution probability matrix as $P(x_a)^T = G - H$ with columns g_i, h_i given by

$$G = [g_1 \;\; g_2 \;\; g_3 \;\; g_4] = \begin{bmatrix} \frac{1}{4} + \frac{3}{4} e^{-\mu x_a} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} + \frac{3}{4} e^{-\mu x_a} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} + \frac{3}{4} e^{-\mu x_a} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} + \frac{3}{4} e^{-\mu x_a} \end{bmatrix}^T$$

and

$$H = [h_1 \;\; h_2 \;\; h_3 \;\; h_4] = \begin{bmatrix} 0 & \frac{1}{4} e^{-\mu x_a} & \frac{1}{4} e^{-\mu x_a} & \frac{1}{4} e^{-\mu x_a} \\ \frac{1}{4} e^{-\mu x_a} & 0 & \frac{1}{4} e^{-\mu x_a} & \frac{1}{4} e^{-\mu x_a} \\ \frac{1}{4} e^{-\mu x_a} & \frac{1}{4} e^{-\mu x_a} & 0 & \frac{1}{4} e^{-\mu x_a} \\ \frac{1}{4} e^{-\mu x_a} & \frac{1}{4} e^{-\mu x_a} & \frac{1}{4} e^{-\mu x_a} & 0 \end{bmatrix}^T.$$

Then we may write

$$L_i^c(x;s,a) = \sum_{j \in \mathcal{N}} p_{ji}(x_a) L_j^a(x;s) = P(t)^T L^a(x;s) = (G - H)(u - v).$$

This has the explicit d.c. decomposition (via Proposition 2.1)

$$\frac{1}{2}\sum_{k=1}^{4}\left[(u_k \oplus g_k)^2 + (v_k \oplus h_k)^2\right] - \frac{1}{2}\sum_{k=1}^{4}\left[(u_k \oplus h_k)^2 + (v_k \oplus g_k)^2\right],$$

where the $\oplus$ operator and the squares stand for Hadamard (componentwise) addition and exponentiation.

Finally, we point out that most nucleotide substitution models in the literature admit closed form formulas for the eigenvalues and the eigenvectors of Q, and hence also for the substitution probabilities $P(t)$. In solving the d.c. program (3.3) (with Algorithm 2.2, for instance), such formulas offer the added computational advantage of circumventing expensive matrix exponentials in function and gradient evaluations.

5. Numerical Results. ML estimates of branch lengths were computed for the first four data sets listed in Table 5.1. Although these data sets represent toy instances in phylogenetic analysis, they serve as a 'proof of concept' validation of the approach developed here.

For each of these data sets, a topology was first generated using PHYLIP [8]. Bound constraints $0.0001 \leq x_i \leq 1.5$ were imposed on all branch lengths x_i in every data set, except in Plant Viroids #2 where the lower bound was 0.001. For each data set, a strictly feasible starting point was chosen at random. Sequences were evolved according to the JC69 model (3.8) with $\mu = 4/3$. The algorithm was terminated when the right hand side in the optimality condition (2.3) was larger than $\epsilon = -0.001$. The computed maximum likelihood values and optimal branch lengths, confirmed by PAML [25], are shown in Figure 5.1 – Figure 5.3.

Data set	n (sequences)	m (sites)
Gene #1, Acetylcholine receptor[a]	3	1368
Influenza A virus[b]	3	890
Plant viroids #1[b]	4	370
Plant viroids #2[b]	5	295
Primates[a]	5	890

[a] PAML distribution [25].
[b] European Bioinformatics Institute website [4].

TABLE 5.1
Details of test data.

Even for a fixed topology and a simple evolutionary model such as JC69, the likelihood is a complicated nonlinear function of the branch lengths (see (A.1) and (A.2) in Appendix A for an example). Although such a function is likely to have multiple local maxima, Fukami and Tateno [9] claimed that the likelihood function has a unique maximizer. However, an error was later discovered by Steel [19], who showed the existence of multiple local maxima, albeit on a contrived example. Rogers and Swofford [18] later argued (based on empirical evidence) that multiple local maxima were unlikely with real genetic data even if the substitution model was erroneous, provided the correct topology was chosen. A subsequent analysis in [2], assisted by MAPLE computations, shows that even with only four taxa and an optimal topology, there exists a wide range of sequence data for which the likelihood function has multiple local maxima.

For the last data set in Table 5.1, we identify two stationary points (of the likelihood function) shown in Figure 5.4 and Figure 5.5; one of these is the global maximum while the other is a saddle point.

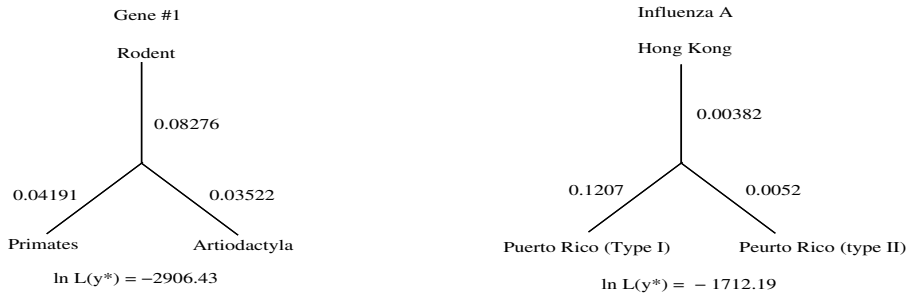

FIG. 5.1. *Gene #1 and Influenza A.*

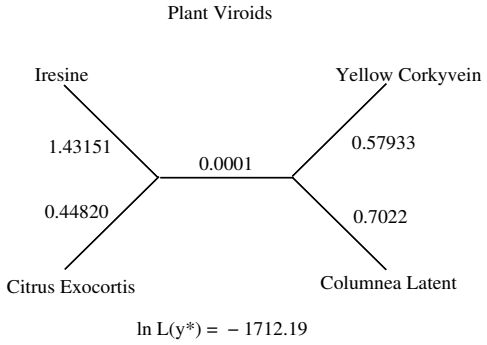

FIG. 5.2. *Plant viroids #1.*

Although the issue of multiple local maxima appears not to have been fully resolved, global optimization techniques are evidently relevant in phylogenetic analysis.

6. Extensions. So far, we have assumed a fixed toplogy τ and focused only on the problem of branch length estimation. As such, this can be considered a bounding procedure in a branch and bound approach to determine the optimal topology. Here we mention some possibilities for incorporating this procedure into a practically successful heuristic, known as *stepwise addition*, for topology optimization. This heuristic, due to Felsenstein [7], is implemented in PHYLIP [8].

Here a particular order of the input sequences is first chosen. The first 3 sequences in this order uniquely determine the topology of a tree T_3 with 3 leaf nodes, and three branches can then be chosen to maximize likelihood. For $k \geq 3$, we now describe how T_{k+1} is constructed from T_k. Given an optimal tree T_k on k sequences (*i.e.* with k leaf nodes), the $k+1^{\text{th}}$ sequence may be added at any one of the $2k-3$ branches in T_k to yield $2k-3$ new trees $T_{k+1}^1, \ldots, T_{k+1}^{2k-3}$, each with $k+1$ leaf nodes and $(2k-3)+2$ branches. Denoting these branch lengths again by the (slightly longer) vector x and the likelihood functions of these trees by $L(x; T_{k+1}^i)$ $(i = 1, \ldots, 2k-3)$, an optimal branch length vector $x^{(i)}$ that maximizes $L(x; T_{k+1}^i)$ may be computed for each $i = 1, \ldots, 2k-3$:

$$x^{(i)} = \arg\max_{x \geq 0} \; L(x; T_{k+1}^i) \quad (i = 1, \ldots, 2k-3). \tag{6.1}$$

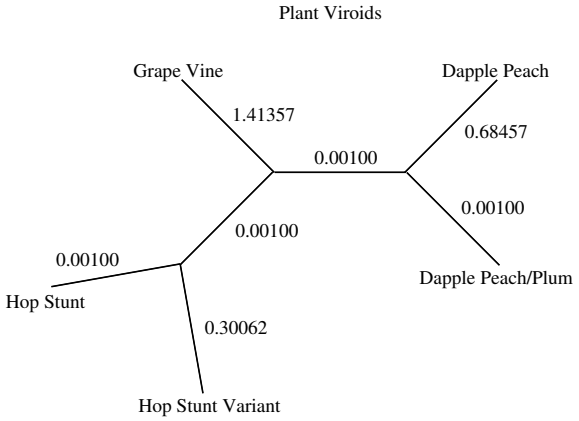

FIG. 5.3. *Plant viroids #2.*

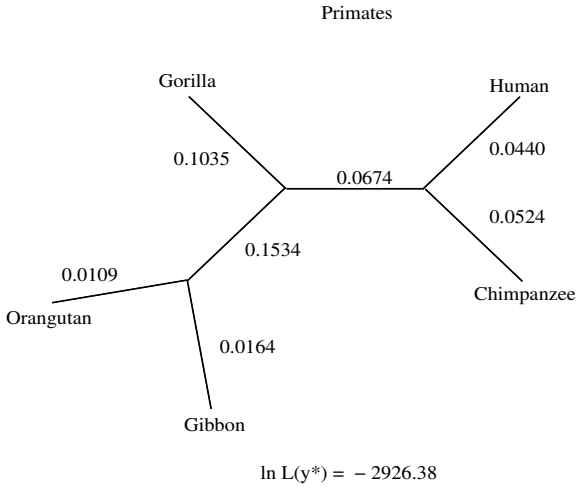

FIG. 5.4. *Primates #1.*

The optimal tree T_{k+1} on $k+1$ sequences is then chosen to be that tree T_{k+1}^i that achieves the largest likelihood:

$$T_{k+1} = \arg\max_{T_{k+1}^i} \; L(x^{(i)}; T_{k+1}^i). \tag{6.2}$$

Thus the tree is grown incrementally until no input sequence remains to be added. (This procedure is usually coupled with other heuristics that locally or globally rearrange subtrees to further explore the set of tree topologies.) Since the topology generated in this manner depends on the original ordering of the input sequences, the whole process is repeated numerous times from different random orderings of the input sequences, finally taking as optimal the tree that achieved the maximum likelihood over all runs.

PHYLIP uses coordinate optimization, solving each of the $2k-3$ problems in (6.1) to local optimality, although only one of them will be selected as the best candidate

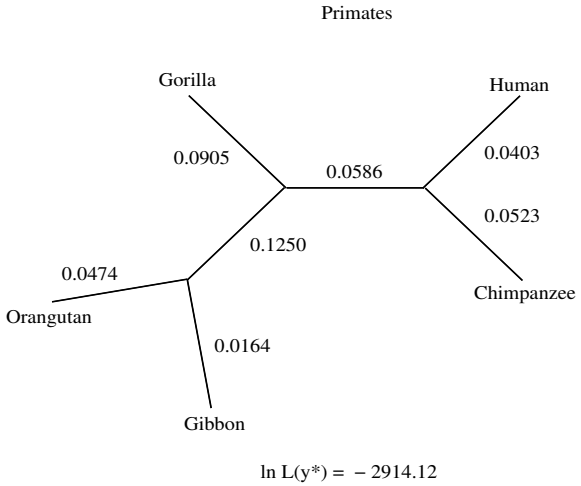

FIG. 5.5. *Primates #2.*

T_{k+1} in (6.2). However, since each $L(x; T^i_{k+1})$ is d.c., we may combine (6.1) into a single d.c. program:

$$T_{k+1} = \arg\max_{T^i_{k+1}} \max_{1\le i\le 2k-3} \left\{ L(x; T^i_{k+1}) \right\}.$$

A similar d.c. formulation, with its concomitant efficiencies, is applicable in *quartet puzzling*, a slightly more involved topology generation heuristic due to [20], but we omit the details.

7. Concluding Remarks. We have presented an approach to compute, under standard assumptions, globally optimal maximum likelihood estimates of branch lengths (divergence times) in a phylogenetic tree of known topology, by exploiting the d.c. structure underlying the likelihood function (Section 3). It is an independent question how closely biological evolution in nature is captured by the assumed probabilistic models in the literature, but within their confines, the globally optimal solutions generated by this approach are arguably superior to the locally optimal solutions computed by the heuristic methods implemented in existing codes. The proposed approach also offers the possibility of incorporating *a priori* biological information that can be encoded as convex constraints. For instance, we may impose upper or lower bounds on the length of a clade, or on the evolutionary distance between two species in the tree, or on ratios of branch lengths, *etc.* While the simple d.c. programming algorithm used here was sufficient for small data sets, larger problems would require a more sophisticated algorithm. One possibility is to embed an effecive local method such as DCA [22, 23] within a branch and bound procedure. A worthwhile effort for future research is to combine such an algorithm with more realistic models (*e.g.* incorporating unknown instantaneous rate parameters, site rate heterogeneity, correlated mutations across sites *etc.*) in an efficient, parallel implementation specially tailored for large scale phylogenetic analysis.

REFERENCES

[1] L. L. Cavalli-Sforza and A. W. F. Edwards, *Phylogenetic analysis: Models and estimation procedures*, Journal of Molecular Evolution, 21 (1967), pp. 550–570.

[2] B. Chor, M. D. Hendy, B. R. Holland, and D. Penny, *Multiple maxima of likelihood in phylogenetic trees: an analytic approach*, Molecular Biology and Evolution, 17 (2000), pp. 1529–1541.

[3] B. Chor and T. Tuller, *Maximum likelihood of evolutionary trees is hard*, in Proceedings of RECOMB 2005, Cambridge, Massachusetts, May 2005.

[4] European Bioinformatics Institute. URL: http://www.ebi.ac.uk.

[5] A. M. F. Rodriguez, J. F. Oliver and J. R. Medina, *The general stochastic model of nucleotide substitutions*, Journal of Theroetical Biology, 142 (1990), pp. 485–501.

[6] J. Felsenstein, *Phylogeny programs*. Available at URL: http://evolution.genetics.washington.edu/phylip/software.html.

[7] ———, *Evolutionary trees from DNA sequences: a maximum likelihood approach*, Journal of Molecular Evolution, 17 (1981), pp. 368–376.

[8] ———, *PHYLIP: Phylogeny Inference Package (Version 3.63)*, Department of Genome Sciences, University of Washington, Seattle, December 2004. Available from: http://evolution.genetics.washington.edu/phylip.html.

[9] K. Fukami and Y. Tateno, *On the uniqueness of the maximum likelihood method for estimating molecular trees: uniqueness of the likelihood point*, Journal of Molecular Evolution, 28 (1989), pp. 460–464.

[10] P. Hartman, *On functions representable as a difference of convex functions*, Pacific Journal of Mathematics, 9 (1959), pp. 707–713.

[11] M. Hasegawa, H. Kishino, and T. Yano, *Dating of the human-ape splitting by a molecular clock of mitchondrial DNA*, Journal of Molecular Evolution, 22 (1985), pp. 160–174.

[12] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*, Kluwer Academic Publishers, Netherlands, second ed., 2000.

[13] R. Horst and N. V. Thoai, *DC programming: overview*, Journal of Optimization Theory and Applications, 103 (1999), pp. 1–41.

[14] J. P. Huelsenbeck, *Performance of phylogenetic methods in simulation*, Systematic Biology, 44 (1995), pp. 17–48.

[15] T. H. Jukes and C. Cantor, *Mammalian Protein Metabolism*, Academic Press, New York, 1969, ch. Evolution of protein molecules, pp. 21–132.

[16] M. Kimura, *A simple model for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences*, Journal of Molecular Evolution, 16 (1980), pp. 111–120.

[17] J. Ponstein, ed., *Generalized differentiability, duality and optimization for problems dealing with differences of convex functions*, no. 256 in Lecture Notes in Economics and Mathematical Systems, New York, June 1984, Springer-Verlag.

[18] J. Rogers and D. Swofford, *Multiple local maxima for likelihoods of phylogenetic trees: a simulation study*, Molecular Biology and Evolution, 16 (1999), pp. 1079–1085.

[19] M. Steel, *The maximum likelihood point for a phylogenetic tree is not unique*, Systematic Biology, 43 (1994), pp. 560–564.

[20] K. Strimmer and A. von Haeseler, *Quartet puzzling: A quartete maximum-likelihood method for reconstructing tree topologies*, Molecular Biology and Evolution, 13 (1996), pp. 964–969.

[21] K. Tamura and M. Nei, *Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees*, Molecular Biology and Evolution, 10 (1993), pp. 512–526.

[22] P. D. Tao and L. T. H. An, *Convex analysis approach to D.C. programming: theory, algorithms and applications*, Acta Mathematica Vietnamica, 22 (1997), pp. 289–355.

[23] ———, *DC (difference of convex) programming. Theory, algorithms, applications: the state-of-the-art*, in Proceedings of the First International Workshop on Global Constrained Optimization and Constraint Satisfaction, Valbonne-Sophia Antipolos (France), October 2002.

[24] H. Tuy, *Convex Analysis and Global Optimization*, Kluwer Acedemic Publishers, Dordrecht, Netherlands, 1998.

[25] Z. Yang, *PAML: Phylogenetic Analysis by Maximum Likelihood — User's Guide (Version 3.14)*, Department of Biology, University College of London, London, UK, September 2004. Available from: http://http://abacus.gene.ucl.ac.uk/software/paml.html.

[26] A. Zharkikh, *Estimation of evolutionary distances between nucleotide sequences*, Journal of Molecular Evolution, 39 (1994), pp. 315–339.

Sequence name	Data
Puerto Rico (Type I)	AGCTCATGAGTACTGCACATGA
Hong Kong	AGCTTCCTGATTAACTTGAGAG
Puerto Rico (Type II)	AGCTCATGAGCACTGTCCGTAG

TABLE A.1
Sequences for the Influenza A data set with column weights.

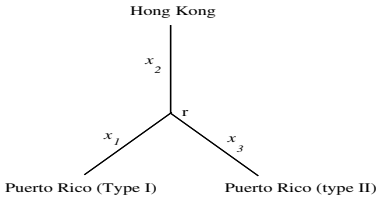

FIG. A.1. *The (unique) topology for the Influenza data set. The node labeled r is chosen as the root node.*

Appendix A. A Simple Example. We give a simple, concrete example of sequence data and the resulting d.c. decomposition of the likelihood function. The sequence data for the *Influenza A* data set used in Section 5 has 890 sites which may be categorized into the 22 patterns shown in Table A.1. The weight of each pattern $w_i \;\; (i = 1, \ldots, 22)$, which is the number of occurrences of that pattern, is shown in the weight vector

$$w = [\, 250 \;\; 186 \;\; 157 \;\; 191 \;\; 13 \;\; 5 \;\; 12 \;\; 4 \;\; 20 \;\; 23 \;\; 2 \;\; 6 \;\; 7 \;\; 4 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 2 \;\; 2 \;\; 1 \;\; 1 \,] .$$

The unique tree on these three taxa is shown in Figure A.1. The likelihood function $L^r(x;1)$ for the first site pattern in Table A.1 has the d.c. decomposition:

$$\begin{aligned}
L^r(x;1) = \;& g(x) - h(x), \quad \text{with} \\
g(x) = \;& \tfrac{13}{128} + \tfrac{3}{8}e^{-x_2} + \tfrac{133}{256}e^{-2x_2} + \tfrac{23}{1024}e^{-2x_1} + \tfrac{111}{256}e^{-3x_2} + \tfrac{1}{64}e^{-x_1} + \\
& \tfrac{1}{8}e^{-x_3} + \tfrac{81}{256}e^{-4x_2} + \tfrac{17}{1024}e^{-3x_1} + \tfrac{1}{1024}e^{-4x_1} + \tfrac{1}{16}e^{-2x_3} + \tfrac{11}{512}e^{-2x_1-x_2} + \\
& \tfrac{81}{1024}e^{-2x_1-2x_2} + \tfrac{3}{512}e^{-x_1-x_2} + \tfrac{9}{128}e^{-2x_1-x_3} + \tfrac{1}{128}e^{-x_1-x_3} + \\
& \tfrac{1}{128}e^{-2x_1-x_3} + \tfrac{9}{1024}e^{-x_1-2x_2} \\
h(x) = \;& \tfrac{1}{128} + \tfrac{3}{8}e^{-2x_2} + \tfrac{81}{256}e^{-4x_2} + \tfrac{13}{1024}e^{-x_1} + \tfrac{131}{256}e^{-x_2} + \tfrac{1}{64}e^{-2x_1} + \\
& \tfrac{1}{2}e^{-x_3} + \tfrac{27}{256}e^{-3x_2} + \tfrac{23}{512}e^{-x_1} + \tfrac{1}{128}e^{-3x_1} + \tfrac{1}{4}e^{-2x_3} + \tfrac{1}{512}e^{-x_1-4x_2} + \\
& \tfrac{243}{1024}e^{-4x_1-4x_2} + \tfrac{3}{256}e^{-2x_1-2x_2} + \tfrac{9}{64}e^{-x_1-x_3} + \tfrac{1}{1024}e^{-2x_1-x_3} + \\
& \tfrac{1}{128}e^{-x_1-x_3} + \tfrac{3}{512}e^{-x_1-4x_2}.
\end{aligned} \tag{A.1}$$

Similar expressions may be calculated for the remaining 21 site patterns. The overall negative log likelihood is a rather complicated function given by the weighted sum

$$-\ln L^r(x) = -\sum_{i=1}^{22} w_i \ln L^r(x;i). \tag{A.2}$$