Generalized Buneman Pruning for Inferring the Most Parsimonious Multi-state Phylogeny

Navodit Misra[1], Guy Blelloch[2], R. Ravi[3], and Russell Schwartz[4]

[1] Department of Physics, Carnegie Mellon University, Pittsburgh, USA
nmisra@andrew.cmu.edu

[2] Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
guyb@cs.cmu.edu

[3] Tepper School of Business, Carnegie Mellon University, Pittsburgh, USA
ravi@cmu.edu

[4] Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, USA
russells@andrew.cmu.edu

Abstract. Accurate reconstruction of phylogenies remains a key challenge in evolutionary biology. Most biologically plausible formulations of the problem are formally NP-hard, with no known efficient solution. The standard in practice are fast heuristic methods that are empirically known to work very well in general, but can yield results arbitrarily far from optimal. Practical exact methods, which yield exponential worst-case running times but generally much better times in practice, provide an important alternative. We report progress in this direction by introducing a provably optimal method for the weighted multi-state maximum parsimony phylogeny problem. The method is based on generalizing the notion of the Buneman graph, a construction key to efficient exact methods for binary sequences, so as to apply to sequences with arbitrary finite numbers of states with arbitrary state transition weights. We implement an integer linear programming (ILP) method for the multi-state problem using this generalized Buneman graph and demonstrate that the resulting method is able to solve data sets that are intractable by prior exact methods in run times comparable with popular heuristics. Our work provides the first method for provably optimal maximum parsimony phylogeny inference that is practical for multi-state data sets of more than a few characters.

1 Introduction

One of the fundamental problems in computational biology is that of inferring evolutionary relationships between a set of observed amino acid sequences or taxa. These evolutionary relationships are commonly represented by a tree (phylogeny) describing the descent of all observed taxa from a common ancestor, a reasonable model provided we are working with sequences over small enough regions or distant enough relationships that we can neglect recombination or other sources of reticulation [1]. Several criteria have been implemented in the literature for inferring phylogenies, of which one of the most popular is maximum

B. Berger (Ed.): RECOMB 2010, LNBI 6044, pp. 369–383, 2010.

parsimony (MP). Maximum parsimony defines the tree(s) with the fewest mutations as the optimum, generally a reasonable assumption for short time-scales or conserved sequences. It is a simple, non-parametric criterion, as opposed to common maximum likelihood models or various popular distance-based methods [2]. Nonetheless, MP is known to be NP-hard [3] and practical implementations of MP are therefore generally based on heuristics which do not guarantee optimal solutions.

For sequences where each site or character is expressed over a set of discrete states, MP is equivalent to finding a minimum Steiner tree displaying the input taxa. For example, general DNA sequences can be expressed as strings of four nucleotide states and proteins as strings of 20 amino acid states. Recently, Sridhar *et al.* [4] used integer linear programming to efficiently find global optima for the special case of sequences with binary characters, which are important when analyzing single nucleotide polymorphism (SNP) data. The solution was made tractable in practice in large part by a pruning scheme proposed by Buneman and extended by others [5,6,7]. The so-called Buneman graph $\mathcal{B}$ for a given set of observed strings is an induced sub-graph of the complete graph $\mathcal{G}$ (whose nodes represent all possible strings of mutations) such that $\mathcal{B} \subseteq \mathcal{G}$ still contains all distinct minimum Steiner trees for the observed data. By finding the Buneman graph, one can often greatly restrict the space of possible solutions to the Steiner tree problem. While there have been prior generalizations of the Buneman graph to non-binary characters [8,9], they do not provide any comparable guarantees usable for accelerating Steiner tree inference.

In this paper, we provide a new generalization of the definition of Buneman graph for any finite number of states that guarantees the resulting graph will contain all distinct minimum Steiner trees of the multi-state input set. Further, we allow transitions between different states to have independent weights. We then utilize the integer linear programming techniques developed in [4] to find provably optimal solutions to the multi-state MP phylogeny problem. We validate our method on four specific data sets chosen to exhibit different levels of difficulty: a set of nucleotide sequences from *Oryza rufipogon* [10], a set of human mt-DNA sequences representing prehistoric settlements in Australia [11], a set of HIV-1 reverse transcriptase amino acid sequences and, finally, a 500 taxa human mitochondrial DNA data set. We further compare the performance of our method, in terms of both accuracy and efficiency, with leading heuristics, `PAUP*` [12] and the `pars` program of PHYLIP [13], showing our method to yield comparable and often far superior run times on non-trivial data sets.

2 Methods

2.1 Notation and Background

Let H be an input matrix that specifies a set of N taxa χ, over a set of m characters $C = \{c_1, \ldots c_m\}$ such that H_{ij} represents the j^{th} character of the i^{th} taxon. The taxa of H represent the terminal nodes of the Steiner tree inference. Further, let n_k be the number of admissible states of the k^{th} character c_k. The set of all possible states is the space $\mathcal{S} \equiv \{0, 1, \ldots n_1 - 1\} \bigotimes \ldots \bigotimes \{0, 1, \ldots n_m - 1\}$. We will represent the i^{th} character of any element $b \in S$, by $(b)_i$. The state space $\mathcal{S}$ can be represented as a graph $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ with the vertex set $V_\mathcal{G} = \mathcal{S}$ and edge set $E_\mathcal{G} = \{(u, v) | u, v \in \mathcal{S}, \sum_{c_p \in C}^{m} \delta[(u)_p, (v)_p] = 1\}$, where $\delta[a, b] = 0$ if $a = b$ and 1 otherwise. Furthermore, let $\boldsymbol{\alpha} = \{\alpha_p | c_p \in C\}$ be a set of weights, such that $\alpha_p[i, j]$ represents an edge length for a transition between states $i, j \in \{0, \ldots n_p - 1\}$ for character c_p. We will assume that these lengths are positive (states that share zero edge length are indistinguishable), symmetric in i, j and satisfy the triangle inequality.

$$\alpha_p[i, j] + \alpha_p[j, k] \geq \alpha_p[i, k] \quad \forall \quad i, j, k \in \{0, \ldots n_p - 1\} \tag{1}$$

Non-negativity and symmetry are basic properties for any reasonable definition of length. If a particular triplet of states (say i, j, k) does not satisfy the triangle inequality in equation 1, we can set $\alpha_p[i, k] = \alpha_p[i, j] + \alpha_p[j, k]$ and still ensure that the shortest path connecting any set of states remains the same. We can now define a distance d_α over $\mathcal{G}$, such that for any two elements $u, v \in V_\mathcal{G}$

$$d_{\boldsymbol{\alpha}}[u, v] \equiv \sum_{p \in C}^{m} \alpha_p[(u)_p, (v)_p] \tag{2}$$

Given any subgraph $K = (V_K, E_K)$ of $\mathcal{G}$, we can define the length of K to be the sum of the lengths of all the edges $L(K) \equiv \sum_{(u,v) \in E_K} d_{\boldsymbol{\alpha}}[u, v]$. The maximum parsimony phylogeny problem for χ is equivalent to constructing the minimum Steiner tree T_* displaying the set of all specified taxa χ, i.e., any tree $T_*(V_*, E_*)$ such that $\chi \subseteq V_*$ and $L(T_*)$ is minimum. Note that T_* need not be unique.

2.2 Pre-processing

Before we construct the generalized Buneman graph corresponding to an input, we perform a basic pre-processing of the data. The set of taxa in the input H might not all be distinct over the length of sequence represented in H. These correspond to identical rows in H and are eliminated. Similarly, characters that do not mutate for any taxa do not affect the true phylogeny and can be removed. Furthermore, if two characters are expressed identically in χ (modulo a relabeling of the states), we will represent them by a single character with each edge length replaced by the sum of the edge lengths of the individual characters. In case there are n such non-distinct characters, one of them is given edge lengths equal to the sum of the corresponding edges in each of the n characters and the rest are discarded. These basic pre-processing steps are often useful in considerably reducing the size of input.

2.3 Buneman Graph

The Buneman graph was introduced as a pruning of the complete graph for the special case of binary valued characters. For this special case it is useful to introduce the notion of binary splits $c_p(0)|c_p(1)$ for each character $c_p \in C$, which partition the set of taxa χ into two sets $c_p(0)$ and $c_p(1)$ corresponding to the value expressed by c_p. Each of these sets is called a block of c_p. Each vertex of the Buneman graph $\mathcal{B}$ can be represented by an m-tuple of blocks $[c_1(i_1), c_2(i_2), \ldots, c_m(i_m)]$, where $i_j = 0$ or 1, for $j \in \{1, 2, \ldots m\}$. To construct the Buneman graph, a rule is defined for discarding/retaining the subset of vertices contained in each pair of overlapping blocks $[c_p(i_p), c_q(i_q)]$ for each pair of characters $(c_p, c_q) \in C \times C$. All vertices which satisfy $c_p(i_p) \cap c_q(i_q) = \emptyset$ for any pair of characters (c_p, c_q) can be eliminated, while those for which $c_p(i_p) \cap c_q(i_q) \neq \emptyset$ for all $[c_p(i_p), c_q(i_q)]$ are retained. Buneman previously established for the binary case that the retained vertex set will contain all terminal and Steiner nodes of all distinct minimum length Steiner trees.

We extend this prior result to the weighted multi-state case by presenting an algorithm analogous to the binary case to construct a graph with these properties.

2.4 Algorithm for Constructing the Generalized Buneman Graph

Briefly, the algorithm looks at the input matrix projected onto each distinct pair of characters p, q and constructs a $n_p \times n_q$ matrix $C(p, q)$, where the $i \times j^{th}$ element $C(p, q)_{ij}$ is 1 only if there is at least one taxon t such that $(t)_p = i$ and $(t)_q = j$ and zero otherwise. The algorithm then implements a rule for each such pair of characters p, q that allows us to enumerate the possible states of those characters in any optimal Steiner tree. For clarity, we will assume that each state for each character is expressed in at least one input taxon, since states that are not present in any taxa cannot be present in a minimum length tree because of the triangle inequality. The rule is defined by a $n_p \times n_q$ matrix $R(p, q)$ determined by the following algorithm:

1. $R(p, q)_{ij} \leftarrow C(p, q)_{ij}$ for all $i \in \{0, 1, \ldots n_p - 1\}$ and $j \in \{0, 1, \ldots n_q - 1\}$.
2. If all non-zero entries in $C(p, q)$ are contained in the set of elements
$$(\cup_k C(p, q)_{ik}) \bigcup (\cup_k C(p, q)_{kj})$$
for a unique pair $i \in \{0, 1, \ldots n_p - 1\}$ and $j \in \{0, 1, \ldots n_q - 1\}$ then $R(p, q)_{xy} \leftarrow 1$ for all x, y such that either $x = i$ or $y = j$ (See Fig 1 where this pair of states are denoted i_{pq} and i_{qp}.)
3. If the condition in step 2 is not satisfied then set $R(p, q)_{ij} \leftarrow 1$ for all i, j.

This set of rules $\{R\}$ then defines a subgraph $B_{pq} \subseteq \mathcal{G}$ for each pair of characters p, q, such that any vertex $v \in B_{pq}$ if and only if $R(p, q)_{(v)_p (v)_q} = 1$. The intersection of these subgraphs $\mathcal{B} = \cap_{c_p, c_q \in C} B_{pq}$ then gives the generalized Buneman graph for χ given any set of distance metrics $\boldsymbol{\alpha} = \{\alpha_p | c_p \in C\}$. Note that the Buneman graph of any subset of χ is a subset of $\mathcal{B}$. It is easily verified that for binary characters, our algorithm yields the standard Buneman graph.

The remainder of this paper will make two contributions. First, it will show that the generalized Buneman graph $\mathcal{B}$ defined above contains all minimum

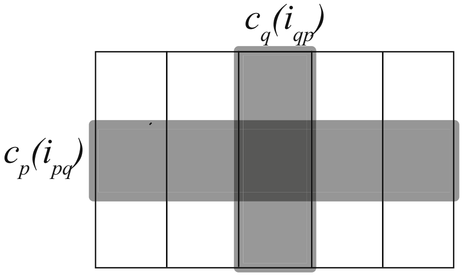


Fig. 1. An example of the generalized Buneman pruning condition. If all taxa in χ are present in the shaded region, vertices in all other blocks can be discarded.

Steiner trees for the input taxa χ. This will in turn establish that restricting the search space for minimum Steiner trees to $\mathcal{B}$ will not affect the correctness of the search. The paper will then empirically demonstrate the value of these methods to efficiently finding minimum Steiner trees in practice.

Before we prove that all Steiner minimum trees connecting the taxa are displayed in $\mathcal{B}$, we need to introduce the notion of a *neighborhood decomposition.* Suppose we are given any tree $T(V, E)$ displaying the set of taxa χ. We will contract each degree-two Steiner node (i.e., any node that is not present in χ) and replace its two incident edges by a single weighted edge. Such trees are called *X-Trees* [14]. Each X-Tree can be uniquely decomposed into its *phylogenetic X-Tree* components, which are maximal subtrees whose leaves are taxa. Formally, each phylogenetic X-Tree $P(\psi)$ consists of a set of taxa $\psi \subseteq \chi$ and a tree displaying them, such that there is a bijection or labeling $\eta : l_P \rightarrow \psi$ between elements of ψ and the set of leaves $l_P \in P(\psi)$ [14] (Fig 2) . All vertices in $P(\psi)$ with degree 3 or higher will be called *branch points.* From now on we will assume that given any input tree, such a decomposition has already been performed (Fig 2). Two phylogenetic X-Trees $P(\psi)$ and $P'(\psi)$ are considered *equivalent* if they have identical length and the same tree topology. By identical tree topology, we mean there is a bijection between the edge set of the two trees, such that removing any edge and its image partitions the leaves into identical bi-partitions. We define two trees to be *neighborhood distinct* if after neighborhood decomposition they differ in at least one phylogenetic X-Tree component. We define a labeling of the phylogenetic X-Tree as an injective map $\Gamma : P \rightarrow \mathcal{G}$ between the vertices of $P(\psi)$ and those of the graph $\mathcal{G}$ such that Γ_u represents the character string for the image of vertex u in $\mathcal{G}$. Since leaf labels are fixed to be the character strings representing the corresponding taxa, $\Gamma_t = \eta_t \in \psi$ for any leaf $t \in l_P$. Identical phylogenetic X-Trees can, however, differ in the labels Γ_u of internal branch points $u \in P \setminus l_P$.

We will use a generalization of the Fitch-Hartigan algorithm to weighted parsimony proposed by Erdos and Szekely [15,16]. The algorithm uses a similar forward pass/backward pass technique to compute an optimal labeling for any phylogenetic X-Tree $T(\psi)$. Arbitrarily root the tree $T(\psi)$ at some taxon ζ and

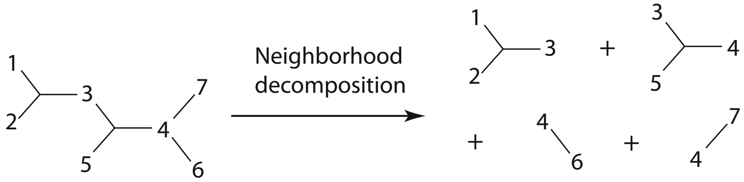


Fig. 2. An input tree and its phylogenetic X-Tree components,with taxa labelled by integers

starting with the leaves compute the minimum length $minL(\Gamma_b, T_b)$ of any labeling of the subtree T_b consisting of the vertex b and its descendants, where the root b is labeled Γ_b as follows.

1. If Γ_b labels a leaf $\eta_b \in \psi$, $minL(\Gamma_b = \eta_b, T_b) = 0$ and ∞ otherwise.
2. If b has k children $D_b = \{v_1, \dots v_k\}$, and T_v is the subtree consisting of $v \in D_b$ and its descendants,

$$minL(\Gamma_b, T_b) = \sum_{v \in D_b} \min_{\Gamma_v} \{minL(\Gamma_v, T_v) + d_{\boldsymbol{\alpha}}[\Gamma_b, \Gamma_v]\} \qquad (3)$$

   where the minimum is to be taken over all possible labels Γ_v for each character and for each child $v \in D_b$.

The optimal labeling of $T(\psi)$ is one which minimizes the length at the root: $L(T) = minL(\eta_\zeta, T_\zeta)$. Labels for each descendant are inferred in a backward pass from the root to the leaves and using equation 3. Note that the minimum length of a tree is just the sum of minimum lengths for each character, i.e., $minL(\Gamma_b, T_b) = \sum_{c_s \in C} minL(\Gamma_b, T_b)^{(s)}$, where $minL(\Gamma_b, T_b)^{(s)}$ is the minimum cost of tree T_b rooted at b for character c_s.

Briefly, our proof is structured as follows: Given any phylogenetic X-Tree $T(\psi)$ labeling (typically denoted Γ below), we will show that the generalized Buneman pruning algorithm for each pair of characters (c_p, c_q) defines a subgraph B_{pq} which contains at least one possible labeling of no higher cost (typically denoted Φ below) for $T(\psi)$. We will then show that the intersection of these subgraphs $\mathcal{B} = \cap_{p \neq q} B_{pq}$ thus contains an optimal labeling for $T(\psi)$.

If the pruning condition in step 2 of the algorithm that defines the Buneman graph is not implemented for the pair of characters (c_p, c_q), then $B_{pq} = \mathcal{G}$ and all labels are necessarily inside B_{pq}. We prove the following lemma for the case when the pruning condition is satisfied, ie., there exist unique states i_{pq} of c_p and i_{qp} of c_q, such that each element in the set of leaves $l_T = \{t \in T(\psi) | \eta_t \in \psi\}$ either has $(\eta_t)_p = i_{pq}$ or $(\eta_t)_q = i_{qp}$ or both. Each time we relabel vertices, we will keep all characters except c_p and c_q fixed. To economize our notation, we will represent the sum of costs in c_p and c_q of the tree T labeled by Γ, which has some branch point b as the root, simply by writing $L(\Gamma, T) = L(\Gamma, T)^{(p)} + L(\Gamma, T)^{(q)}$. We use the notation $\Gamma_x = [(\Gamma_x)_p, (\Gamma_x)_q]$ to represent the label for a vertex x and suppress the state of all other characters.

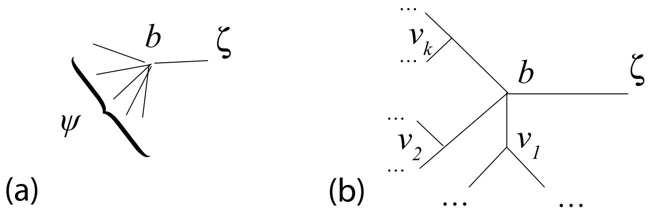


Fig. 3. (a) The base case of a degree $|\psi|$ star that can be attached to a parent vertex ζ in the Erdos-Szekely algorithm. (b) $T(\psi)$ for the general case (see Lemma 1).

Lemma 1. *Given any phylogenetic X-Tree $T(\psi)$ with $\psi \subseteq B_{pq}$, and a labeling Γ, such that an internal branch point $b \in T \setminus l_T$ is labeled outside B_{pq}, i.e., $\Gamma_b \notin B_{pq}$, there exists an alternate labeling Φ of $T(\psi)$ inside B_{pq} such that*

1. *either $L(\Gamma, T) \geq L(\Phi, T) + d_{\boldsymbol{\alpha}}[\Gamma_b, \Phi_b]$, or —*
2. *$L(\Gamma, T) \geq L(\Phi, T)$ for each of the following choices: $\Phi_b = [i_{pq}, i_{qp}]$ or $[i_{pq}, (\Gamma_b)_q]$ or $[(\Gamma_b)_p, i_{qp}]$, and $\Phi_v = \Gamma_v$ for all $v \neq b$. We will call a tree that satisfies this second condition a (c_p, c_q)-Tree*

Proof. We will use induction on the number of internal branch points outside B_{pq} to prove the claim. Without loss of generality we can consider all branch points of $T(\psi)$ to be labeled outside B_{pq}. If some branch points are labeled inside B_{pq} then they can be treated as leaves of smaller X-Tree(s) that have all branch points outside B_{pq}. This is similar to the neighborhood decomposition we performed earlier for those branch points that were present in the set of input taxa. The set of branch points is then the set $T \setminus l_T = \{u \in T | \Gamma_u \notin B_{pq}\}$.

For the base case assume all the leaves are joined at a single branch point b to form a star of degree $|\psi|$ (see Fig. 3(a) without the root ζ). We can group the leaves into three sets:

1. $I = \{\eta_u = [i_{pq}, y_u] | y_u \neq i_{qp}, \eta_u \in \psi\}$
2. $II = \{\eta_v = [x_v, i_{qp}] | x_v \neq i_{pq}, \eta_v \in \psi\}$
3. $III = \{\eta_w = [i_{pq}, i_{qp}] | \eta_w \in \psi\}$

The cost of the tree for c_p and c_q, with branch point $\Gamma_b = [x, y]$, is

$$L(\Gamma,T)^{(p)} + L(\Gamma,T)^{(q)} = \sum_{u \in I} (\alpha_p[x, i_{pq}] + \alpha_q[y, y_u]) + \sum_{v \in II} (\alpha_p[x, x_v] + \alpha_q[y, i_{qp}]) + \sum_{w \in III} (\alpha_p[x, i_{pq}] + \alpha_q[y, i_{qp}]) \qquad (4)$$

The only way for $L(\Gamma,T)^{(p)} + L(\Gamma_b,T)^{(q)}$ to be minimum with $x \neq i_{pq}$ and $y \neq i_{qp}$, is if $III = \emptyset$ and $|I| = |II|$. For contradiction, suppose $|I| + |III| > |II|$. We could then define a labeling Φ identical to Γ over all characters, except $\Phi_b = [i_{pq}, y]$, such that $d_{\boldsymbol{\alpha}}[\Gamma_b, \Phi_b] = \alpha_p[\Gamma_b, \Phi_b]$. We could then reduce the length, since

$$
\begin{aligned}
L(\Gamma, T)^{(p)} &= \sum_{u \in I} \alpha_p[x, i_{pq}] + \sum_{v \in II} \alpha_p[x, x_v] + \sum_{w \in III} \alpha_p[x, i_{pq}] \\
&\geq \alpha_p[x, i_{pq}] + \sum_{v \in II} (\alpha_p[x, x_v] + \alpha_p[x, i_{pq}]) \\
&\geq \alpha_p[x, i_{pq}] + \sum_{v \in II} \alpha_p[i_{pq}, x_v] = L(\Phi, T)^{(p)} + d_{\boldsymbol{\alpha}}[\Gamma_b, \Phi_b] \qquad (5)
\end{aligned}
$$

where the last inequality follows from the triangle inequality. Similarly, if $|II| + |III| > |I|$, we could define $\Phi_b = [x, i_{qp}]$ and arrive at $L(\Gamma, T)^{(q)} \geq L(\Phi, T)^{(q)} + d_{\boldsymbol{\alpha}}[\Gamma_b, \Phi_b]$.

On the other hand if $|I| = |II|$ and $III = \emptyset$ setting $\Phi_b = [i_{pq}, y]$ or $\Phi_b = [x, i_{qp}]$ or $\Phi_b = [i_{pq}, i_{qp}]$ all achieve a length no more than $L(\Gamma, T)^{(p)} + L(\Gamma, T)^{(q)}$. Therefore, this is a (c_p, c_q)-Tree. This proves the base case for our proposition.

We will now assume that the claim is true for all trees with n branch points or less. Suppose we have a labeled tree $T(\psi)$ with $n+1$ branch points which are all outside B_{pq}. Let $D_b = \{v_1, \ldots v_k\}$ be the children of a branch point b in $T(\psi)$ and $\{T_1, \ldots T_k\}$ be the subtrees of each $v \in D_b$ and their descendants. Note that some of these descendants may be leaves. Since $T(\psi)$ has at least two branch points, one of its descendants (say v_1) must be a branch point (Fig 3(b)). Let $T_b = T \setminus T_1$ be the subtree consisting of b and all its other descendants. For clarity we will use the notation $\Gamma_b = [x_b, y_b]$ and $\Gamma_{v_1} = [x_1, y_1]$. This implies,

$$
\begin{aligned}
L(\Gamma, T) &= L(\Gamma, T_b) + L(\Gamma, T_1) + d_{\boldsymbol{\alpha}}[\Gamma_b, \Gamma_{v_1}] \\
&= L(\Gamma, T_b) + L(\Gamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \qquad (6)
\end{aligned}
$$

There are four possibilities.

1. Both T_b and T_1 are (c_p, c_q)-Trees with n or less branch points - In this case, by induction, both T_b and T_1 can be relabeled with Φ_b and Φ_{v_1} of the form $[i_{pq}, i_{qp}]$. Since the cost in c_p and c_q of the edge (b, v_1) is now zero, we have an optimal labeling of $T(\psi)$ within B_{pq} and $L(\Gamma, T) \geq L(\Phi, T)$. Note that each of the choices of the form $[i_{pq}, y_1]$ or $[x_1, i_{pq}]$ for relabeling of b also satisfy property 2 of the claim. Therefore, this is a (c_p, c_q)-Tree.
2. T_b is a (c_p, c_q)-Tree, but T_1 is not. Therefore, there is a labeling Φ of T_1 with either $\Phi_{v_1} = [i_{pq}, y_1]$ and/or $\Phi_{v_1} = [x_1, i_{pq}]$ such that

$$
L(\Gamma, T_1) \geq L(\Phi, T_1) + d_{\boldsymbol{\alpha}}[\Gamma_{v_1}, \Phi_{v_1}] \qquad (7)
$$

   Let us assume for concreteness that $\Phi_{v_1} = [i_{pq}, y_1]$. It will become clear that the argument works for the other possible choices. Since, T_b is a (c_p, c_q)-Tree, by induction, we can choose a labeling of T_b with $\Phi_b = [i_{pq}, y_b]$, such that $L(\Gamma, T_b) \geq L(\Phi, T_b)$. This gives

$$\begin{aligned} L(\varPhi, T) &= L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varPhi_b, \varPhi_{v_1}] \\ &= L(\varPhi, T_b) + L(\varPhi, T_1) + \alpha_q[y_b, y_1] \end{aligned} \tag{8}$$

Comparing the previous two equations with equation 6, we get,

$$\begin{aligned} L(\varGamma, T) &= L(\varGamma, T_b) + L(\varGamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varGamma_{v_1}, \varPhi_{v_1}] + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &= L(\varPhi, T_b) + L(\varPhi, T_1) + \alpha_p[x_1, i_{pq}] + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\varPhi, T_b) + L(\varPhi, T_1) + \alpha_p[x_b, i_{pq}] + \alpha_q[y_b, y_1] \\ &= L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b] + d_{\boldsymbol{\alpha}}[\varPhi_b, \varPhi_{v_1}] \\ &= L(\varPhi, T) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b] \end{aligned} \tag{9}$$

which satisfies the first possibility of the claim. It should be clear that if $\varPhi_{v_1} = [x_1, i_{qp}]$ then the choice $\varPhi_b = [x_b, i_{qp}]$ would give an identical bound.

3. T_1 is a (c_p, c_q)-Tree, but T_b is not. This case is similar to the previous one. Since T_b has less than n branch points, which are all outside B_{pq}, and it is not a (c_p, c_q)-Tree, we have from induction a labeling $\varPhi$ of T_b with either $\varPhi_b = [i_{pq}, y_b]$ and/or $\varPhi_b = [x_b, i_{pq}]$ such that

$$L(\varGamma, T_b) \geq L(\varPhi, T_b) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b] \tag{10}$$

As before, let us assume $\varPhi_b = [i_{pq}, y_b]$ for concreteness. Since T_1 is a (c_p, c_q)-Tree, we can choose a labeling with $\varPhi_{v_1} = [i_{pq}, y_1]$ such that $L(\varGamma, T_1) \geq L(\varPhi, T_1)$. This gives,

$$\begin{aligned} L(\varPhi, T) &= L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varPhi_b, \varPhi_{v_1}] \\ &= L(\varPhi, T_b) + L(\varPhi, T_1) + \alpha_q[y_b, y_1] \end{aligned} \tag{11}$$

Comparing the previous two equations with equation 6, we get,

$$\begin{aligned} L(\varGamma, T) &= L(\varGamma, T_b) + L(\varGamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b] + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b] + d_{\boldsymbol{\alpha}}[\varPhi_b, \varPhi_{v_1}] \\ &= L(\varPhi, T) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b] \end{aligned} \tag{12}$$

An identical argument carries through if $\varPhi_b = [x_b, i_{qp}]$.

4. Neither T_1 or T_b are (c_p, c_q)-Trees. It follows from induction that there is a labeling $\varPhi$ such that $L(\varGamma, T_b) \geq L(\varPhi, T_b) + d_{\boldsymbol{\alpha}}[\varGamma_b, \varPhi_b]$ and $L(\varGamma, T_1) \geq L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varGamma_{v_1}, \varPhi_{v_1}]$. There are two possibilities in this case.
   (a) ($\varPhi_b = [i_{pq}, y_b]$ and $\varPhi_{v_1} = [i_{pq}, y_1]$) or ($\varPhi_b = [x_b, i_{qp}]$ and $\varPhi_{v_1} = [x_1, i_{qp}]$). As before, we will prove the claim for the former possibility while the later case can be proved by an identical argument.

$$\begin{aligned} L(\varPhi, T) &= L(\varPhi, T_b) + L(\varPhi, T_1) + d_{\boldsymbol{\alpha}}[\varPhi_b, \varPhi_{v_1}] \\ &= L(\varPhi, T_b) + L(\varPhi, T_1) + \alpha_q[y_b, y_1] \end{aligned} \tag{13}$$

$$\begin{aligned} L(\Gamma,T) &= L(\Gamma,T_b) + L(\Gamma,T_1) + \alpha_p[x_b,x_1] + \alpha_q[y_b,y_1] \\ &\geq L(\Phi,T_b) + L(\Phi,T_1) + d_{\boldsymbol{\alpha}}[\Gamma_b,\Phi_b] + d_{\boldsymbol{\alpha}}[\Gamma_{v_1},\Phi_{v_1}] \\ &+ \alpha_p[x_b,x_1] + \alpha_q[y_b,y_1] \\ &\geq L(\Phi,T_b) + L(\Phi,T_1) + d_{\boldsymbol{\alpha}}[\Gamma_b,\Phi_b] + \alpha_q[y_b,y_1] \\ &= L(\Phi,T_b) + L(\Phi,T_1) + d_{\boldsymbol{\alpha}}[\Gamma_b,\Phi_b] + d_{\boldsymbol{\alpha}}[\Phi_b,\Phi_{v_1}] \\ &= L(\Phi,T) + d_{\boldsymbol{\alpha}}[\Gamma_b,\Phi_b] \end{aligned} \tag{14}$$

This also satisfies the claim. The proof for $\Phi_b = [x_b, i_{qp}]$ and $\Phi_{v_1} = [x_1, i_{qp}]$ is identical.

(b) ($\Phi_b = [i_{pq}, y_b]$ and $\Phi_{v_1} = [x_1, i_{qp}]$) or ($\Phi_b = [x_b, i_{qp}]$ and $\Phi_{v_1} = [i_{pq}, y_1]$). As before, we show the calculation for the former possibility. In this case

$$\begin{aligned} L(\Phi,T) &= L(\Phi,T_b) + L(\Phi,T_1) + d_{\boldsymbol{\alpha}}[\Phi_b,\Phi_{v_1}] \\ &= L(\Phi,T_b) + L(\Phi,T_1) + \alpha_p[x_b,i_{pq}] + \alpha_q[i_{qp},y_1] \end{aligned} \tag{15}$$

Combining this with equation 6 we get,

$$\begin{aligned} L(\Gamma,T) &= L(\Gamma,T_b) + L(\Gamma,T_1) + \alpha_p[x_b,x_1] + \alpha_q[y_b,y_1] \\ &\geq L(\Phi,T_b) + L(\Phi,T_1) + d_{\boldsymbol{\alpha}}[\Gamma_b,\Phi_b] + d_{\boldsymbol{\alpha}}[\Gamma_{v_1},\Phi_{v_1}] \\ &+ \alpha_p[x_b,x_1] + \alpha_q[y_b,y_1] \\ &= L(\Phi,T_b) + L(\Phi,T_1) + \alpha_p[x_b,i_{pq}] + \alpha_q[i_{qp},y_1] \\ &+ \alpha_p[x_b,x_1] + \alpha_q[y_b,y_1] \\ &\geq L(\Phi,T_b) + L(\Phi,T_1) + \alpha_p[x_b,i_{pq}] + \alpha_q[i_{qp},y_1] \\ &= L(\Phi,T_b) + L(\Phi,T_1) + d_{\boldsymbol{\alpha}}[\Phi_b,\Phi_{v_1}] = L(\Phi_b,T) \end{aligned} \tag{16}$$

But if we now relabel b and v_1 with $\tilde{\Phi}_{v_1} = [i_{pq}, i_{qp}]$ and $\tilde{\Phi}_b = [i_{pq}, i_{qp}]$ while $\tilde{\Phi}_v = \Phi_v$ for all other v, we get $L(\Phi,T_1) + \alpha_q[y_1, i_{qp}] \geq L(\tilde{\Phi}_{v_1}, T_1)$ and $L(\Phi,T_b) + \alpha_p[x_b, i_{pq}] \geq L(\tilde{\Phi}, T_b)$. This immediately gives,

$$\begin{aligned} L(\tilde{\Phi},T) &= L(\tilde{\Phi},T_b) + L(\tilde{\Phi},T_1) + d_{\boldsymbol{\alpha}}[\tilde{\Phi}_b,\tilde{\Phi}_{v_1}] \\ &\geq L(\Phi,T) \geq L(\Gamma,T) \end{aligned} \tag{17}$$

Identical arguments work for the choices $\tilde{\Phi}_{v_1} = [x_1, i_{qp}]$ and $\tilde{\Phi}_b = [x_b, i_{qp}]$.

This proves that if either of the two possibilities claimed are always true for an X-Tree with n branch points or less then they are also true for a tree with $n+1$ branch points. The proof for arbitrary n follows from induction.

□

Corollary 1. *Given a minimum length phylogenetic X-Tree $T(\psi)$ there is an optimal labeling for each branch point within $\mathcal{B}$.*

Proof. Lemma 1 establishes that for any minimum Steiner tree labeled by Γ and any branch point $b \in T$ such that $\Gamma_b \notin B_{pq}$, an alternative optimal labeling Φ exists such that Φ_b is inside the union of blocks

$$\Lambda(\Gamma_b, p, q) \equiv [c_p(i_{pq})c_q(i_{qp})] \cup [c_p(i_{pq})c_q((\Gamma_b)_q)] \cup [c_p((\Gamma_b)_p)c_q(i_{qp})]$$

If we root the tree at b, the new optimal labeling for all its descendants is inferred in a backward pass of the Erdos-Szekely algorithm. This ensures that each branch point in a minimum length phylogenetic X-Tree is labeled inside B_{pq}. Let $S_b = \cap_{B_{pq} \neq \mathcal{G}} \Lambda(\Gamma_b, p, q) \subseteq \mathcal{B}$, where the intersection is taken over all pair of characters for which the pruning condition is satisfied. It follows from Lemma 1 that S_b also contains an alternate optimal labeling of $T(\psi)$. Note that S_b is a non-empty subset of $\mathcal{B}$. This must be true because given a character pair c_p, c_q, each union of blocks contains at least one taxon and so the rule matrix $R(p, q)$ that defines the Buneman graph must have ones for each of these blocks. Therefore each element in S_b represents a distinct vertex of the Buneman graph. □

As argued before, any minimum Steiner tree can be decomposed uniquely into phylogenetic X-Tree components and the previous corollary ensures that each phylogenetic X-Tree can be labeled optimally inside the generalized Buneman graph. It follows that all distinct minimum Steiner trees are contained inside the generalized Buneman graph.

2.5 Integer Linear Program (ILP) Construction

We briefly summarize the ILP flow construction used to find the optimal phylogeny. We convert the generalized Buneman graph into a directed graph by replacing an edge between vertices u and v with two directed edges $(u, v), (v, u)$ each with weight w_{uv} as determined by the distance metric. Each directed edge has a corresponding binary variable $s_{u,v}$ in our ILP. We arbitrarily choose one of the taxa as the root r, which acts as a source for the flow model. The remaining taxa $T \equiv \chi - \{r\}$ correspond to sinks. Next, we set up real-valued flow variables $f^t_{u,v}$, representing the flow along the edge (u, v) that is intended for terminal t. The root r outputs $|T|$ units of flow, one for each terminal. The Steiner tree is the minimum-cost tree satisfying the flow constraints. This ILP was described in [4], and we refer the reader to that paper for further details. The ILP for this construction of the Steiner tree problem is the following:

$$\begin{aligned} \text{Minimize} \sum_{(u,v)\in\mathcal{B}} & w_{uv} s_{u,v} \\ \text{subject to} \sum_{v} (f^t_{u,v} - f^t_{v,u}) &= 0 \quad \forall u \in \mathcal{B} \setminus \{t, r\},\ \forall t \in T \\ \sum_{v} (f^t_{r,v} - f^t_{v,r}) &= 1 \quad \forall t \in T \\ 0 \leq f^t_{u,v} &\leq s_{u,v} \quad \forall (u, v) \in \mathcal{B},\ \forall t \in T \\ s_{u,v} &\in \{0, 1\} \quad \forall (u, v) \in \mathcal{B} \end{aligned} \tag{18}$$

Table 1. Pruning and run time results for the data sets reported

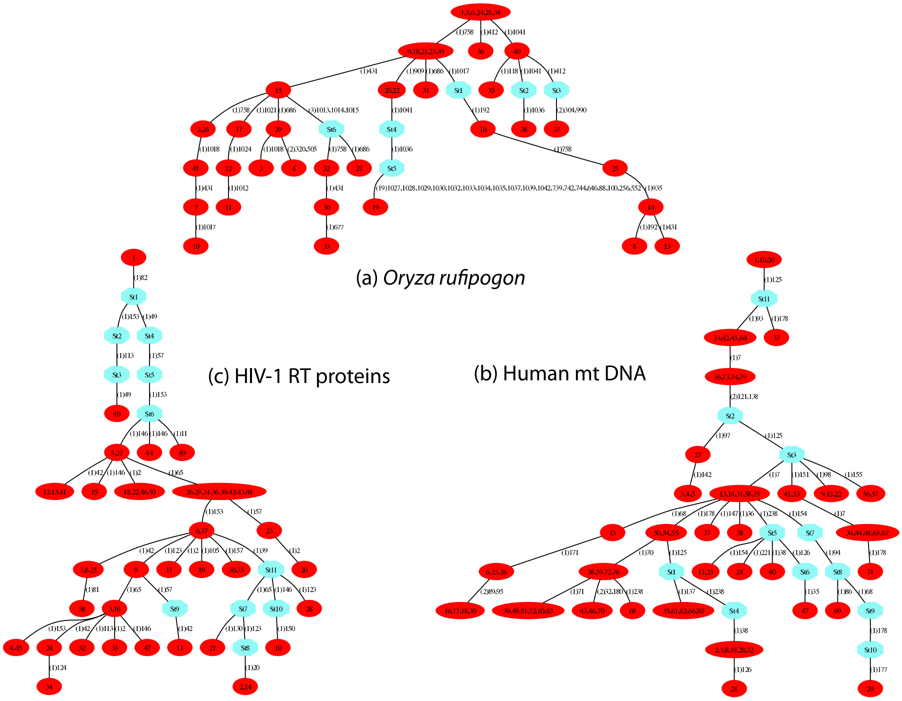
Data	Input	Complete	$\lvert\mathcal{B}\rvert$	ILP		`pars`		`PAUP*`	
	(raw)	graph		length	time	length	time	length	time
O. rufipogon DNA	41×1043	$2^{18} * 3^2$	58	57	0.29s	57	2.57s	57	2.09s
Human mt-DNA	80×245	2^{28}	64	44	0.48s	45	0.56s	44	5.69s
HIV-1 RT protein	50×176	$2^{16} * 3 * 4^2$	297	40	127.5s	42	0.30s	40	3.84s
mt3000	500×3000	$2^{99} * 3^2$	322	177	40s	178	2m37s	177	5h23m
mt5000a	500×5000	$2^{167} * 3^2$	1180	298	5h10m	298	35m49s	298	3h52m
mt5000b	500×5000	$2^{229} * 3^3$	360	312	3m41s	312	57m6s	312	2h40m
mt10000	500×10000	$2^{357} * 3^5$	6006	N. A.	N. A.	637	1h34m	637	1h39m

3 Results

We implemented our generalized Buneman pruning and the ILP in C++. The ILP was solved using the Concert callable library of CPLEX 10.0. We compared the performance of our method with two popular heuristic methods for maximum parsimony phylogeny inference — `pars`, which is part of the freely-available PHYLIP package [13], and `PAUP*` [12], the leading commercial phylogenetics package. We attempted to use PHYLIP's exact branch-and-bound method `DNA penny` for nucleotide sequences, but discontinued the tests when it failed to solve any of the data sets in under 24 hours. In each case, `pars` and `PAUP*` were run with default parameters. We first report results from three moderate-sized data sets selected to provide varying degrees of difficulty: a set of 1,043 sites from a set of 41 sequences of *O. rufipogon* (red rice) [10], 245 positions from a set of 80 human mt-DNA sequences reported by [11], and 176 positions from 50 HIV-1 reverse transcriptase amino acid sequences. The HIV sequences were retrieved by NCBI BLASTP [17] searching for the top 50 best aligned taxa for the query sequence GI 19571541 and default parameters. We then added additional tests on larger data sets all derived from human mitochondrial DNA. The mtDNA data was retrieved from NCBI BLASTN, searching for the 500 best aligned taxa for the query sequence GI 61287976 and default parameters. The complete set of 16,546 characters (after removing indels) was then broken in four windows of varying sizes and characteristics: the first 3,000 characters (mt3000), the first 5,000 characters (mt5000a), the next 5,000 characters (mt5000b), and the first 10,000 characters (mt10000). Table 1 summarizes the results.

For the set of 41 sequences of lhs-1 gene from *O. rufipogon* (red rice) [10], our method pruned the full graph of $2^{18} * 3^2$ nodes (after screening out redundant characters) to 58. Fig 4(a) shows the resulting phylogeny. Both `PAUP*` and `pars` yielded an optimal tree although more slowly than the ILP (2.09 seconds and 2.57 seconds respectively, as opposed to 0.29 seconds).

For the 245-base human mt-DNA sequences, the generalized Buneman pruning was again highly efficient, reducing the state set from 2^{28} after removing redundant sequences to 64. Fig 4(b) shows the phylogeny returned. While `PAUP*` was able to find the optimal phylogeny (although it was again slower at 5.69


(a) *Oryza rufipogon*

(c) HIV-1 RT proteins

(b) Human mt DNA

Fig. 4. Most parsimonious phylogenies (a) lhs-1 gene for *O. rufipogon* [10] (b) Human mt-DNA [11] and (c) HIV-1 RT proteins [17]. Edges are labelled by their lengths in parentheses followed by sites that mutate along that edge. Dark red ovals are input taxa and light blue Steiner nodes.

seconds versus 0.48 seconds), pars yielded a slightly sub-optimal phylogeny (length 45 instead of 44) in a comparable run time (0.56 seconds).

For HIV-1 sequences, our method pruned the full graph of $2^{16} * 3 * 4^2$ possible nodes to a generalized Buneman graph of 297 nodes, allowing solution of the ILP in about two minutes. Fig 4(c) shows an optimal phylogeny for the data. PAUP* was again able to find the optimal phylogeny and in this case was faster than the ILP (3.84 seconds as opposed to 127.5 seconds). pars required a shorter run time of 0.30 seconds, but yielded a sub-optimal tree of length of 42, as opposed to the true minimum of 40.

For the four larger mitochondrial datasets, Buneman pruning was again highly effective in reducing graph size relative to the complete graph, although the ILP approach eventually proves impractical when Buneman graph sizes grows sufficiently large. Two of the data sets yielded Buneman graphs of size below 400, resulting in ILP solutions orders of magnitude faster than the heuristics. mt5000a, however, yielded a Buneman graph of over 1,000 nodes, resulting in an ILP that ran more slowly than the heuristics. mt10000 resulted in a Buneman graph of over 6,000 nodes, leading to an ILP too large to solve. pars was faster than PAUP* in all cases, but PAUP* found optimal solutions for all three instances we can verify while pars found a sub-optimal solution in one instance.

We can thus conclude that the generalized Buneman pruning approach developed here is very effective at reducing problem size, but solving provably to optimality does eventually become impractical for large data sets. Heuristic approaches remain a practical necessity for such cases even though they cannot guarantee, and do not always deliver, optimality. Comparison of PAUP* to pars and the ILP suggests that more aggressive sampling over possible solutions by the heuristics can lead optimality even on very difficult instances but at the cost of generally greatly increased run time on the easy to moderate instances.

4 Discussion

We have presented a new method for finding provably optimal maximum parsimony phylogenies on multi-state characters with weighted state transitions, using integer linear programming. The method builds on a novel generalization of the Buneman graph for characters with arbitrarily large but finite state sets and for arbitrary weight functions on character transitions. Although the method has an exponential worst-case performance, empirical results show that it is fast in practice and is a feasible alternative for data sets as large as a few hundred taxa. While there are many efficient heuristics for recontructing maximum parsimony phylogenies, our results cater to the need for provably exact methods that are fast enough to solve the problem for biologically relevant multi-state data sets. Our work could potentially be extended to include more sophisticated integer programming techniques that have been successful in solving large instances of other hard optimization problems, for instance the recent solution of the 85,900-city traveling salesman problem pla85900 [18]. The theoretical contributions of this paper may also prove useful to work on open problems in multi-state MP phylogenetics, to accelerating methods for related objectives, and to sampling among optimal or near-optimal solutions.

Acknowledgements

NM would like to thank Ming-Chi Tsai for several useful discussions. This work was supported in part by NSF grant #0612099.

References

1. Posada, D., Crandall, K.: Intraspecific gene genealogies: trees grafting into networks. Trends in Ecology and Evolution 16, 37–45 (2001)
2. Felsenstein, J.: Inferring Phylogenies. Sinauer Publications (2004)
3. Foulds, L.R., Graham, R.L.: The Steiner problem in phylogeny is NP-complete. Advances in Applied Mathematics 3, 43–49 (1982)
4. Sridhar, S., Lam, F., Blelloch, G., Ravi, R., Schwartz, R.: Efficiently finding the most parsimonious phylogenetic tree via linear programming. In: Măndoiu, I.I., Zelikovsky, A. (eds.) ISBRA 2007. LNCS (LNBI), vol. 4463, pp. 37–48. Springer, Heidelberg (2007)

5. Buneman, P.: The recovery of trees from measures of dissimilarity. In: Hodson, F., et al. (eds.) Mathematics in the archeological and historical sciences, pp. 387–395 (1971)
6. Barthẽlemy, J.: From copair hypergraphs to median graphs with latent vertices. Discrete Math. 76, 9–28 (1989)
7. Bandelt, H.J., Forster, P., Sykes, B.C., Richards, M.B.: Mitochondrial portraits of human populations using median networks. Genetics 141, 743–753 (1989)
8. Bandelt, H.J., Forster, P., Rohl, A.: Median-joining networks for inferring intraspecific phylogenies. Molecular Biology and Evolution 16, 37–48 (1999)
9. Huber, K.T., Moulton, V.: The relation graph. Discrete Mathematics 244(1-3), 153–166 (2002)
10. Zhou, H.F., Zheng, X.M., Wei, R.X., Second, G., Vaughan, D.A., Ge, S.: Contrasting population genetic structure and gene flow between Oryza rufipogon and Oryza nivara. Theor. Appl. Genet. 117(7), 1181–1189 (2008)
11. Hudjashov, G., Kivisild, T., Underhill, P.A., Endicott, P., Sanchez, J.J., Lin, A.A., Shen, P., Oefner, P., Renfrew, C., Villems, R., Forster, P.: Revealing the prehistoric settlement of Australia by Y chromosome and mtDNA analysis. Proc. Natl. Acad. Sci. USA 104(21), 8726–8730 (2007)
12. Swofford, D.: PAUP* 4.0. Sinauer Assoc. Inc., Sunderland (2009)
13. Felsenstein, J.: PHYLIP (phylogeny Inference package) version 3.6 distributed by author, Department of Genome Sciences, University of Washington, Seattle (2008)
14. Semple, C., Steel, M.: Phylogenetics. Oxford University Press, Oxford (2003)
15. Erdos, P.L., Szekely, L.A.: On weighted multiway cuts in trees. Mathematical Programming 65, 93–105 (1994)
16. Wang, L., Jiang, T., Lawler, L.: Approximation algorithms for tree alignment with a given phylogeny. Algorithmica 16, 302–315 (1996)
17. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402 (1997)
18. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W., Espinoza, D.G., Goycoolea, M., Helsgaun, K.: Certification of an optimal TSP tour through 85,900 cities. Operations Research Letters 37(1), 11–15 (2009)