A Polynomial Time Algorithm for Constructing the Refined Buneman tree

David Bryant

CRM, Université de Montréal, CP 6128 Succeursale Centre-ville, Montréal, Quebec H3C 3J7

Vincent Moulton

FMI, Mid Sweden University, Sundsvall S-851 70, Sweden

Abstract

We present a polynomial time algorithm for computing the refined Buneman tree, thereby making it applicable for tree reconstruction on large data sets. The refined Buneman tree retains many of the desirable properties of its predecessor, the well known Buneman tree, but has the practical advantage that it is typically more refined.

Key words: Phylogenetics, distance-based tree reconstruction, Buneman tree, refined Buneman tree, weighted trees

1 Introduction

Let X be a finite set, and let $\mathcal{D}(X)$ denote the set of distance functions on X, that is, the set of symmetric functions $d : X^2 \to \mathbb{R}$ that are zero on the diagonal. An X-*tree* is a graph theoretical tree $T = (V, E)$ together with a labelling $L : X \to V$ such that all of the vertices in $V - L(X)$ have degree at least three [1,2]. An X-tree together with an edge weighting $w : E \to \mathbb{R}_{>0}$ induces an associated distance function on X: the distance between $x \in X$ and $y \in Y$ is taken to be the sum of the weights $w(e)$ over all edges e in the unique path in T connecting vertices $L(x)$ and $L(y)$. Any distance function arising in this way is called a *tree distance*, and we denote the set of all tree metrics on X by $\mathcal{T}(X)$.

An important problem in phylogenetic analysis is to approximate distances (such as those arising from biomolecular data) by tree metrics, and many various methods have been found for attacking this (see [3,4] for surveys). We investigate this problem by looking for a *tree construction map*, that is, a map $\phi : \mathcal{D}(X) \to \mathcal{D}(X)$, with $\phi(\mathcal{D}(X)) \subseteq \mathcal{T}(X)$, which satisfies the following properties:

(R1) $\phi|_{\mathcal{T}(X)} = Id|_{\mathcal{T}(X)}$.
(R2) The map ϕ is continuous.
(R3) The map ϕ is *homogeneous*, i.e. $\phi(\lambda d) = \lambda\phi(d)$, for $d \in \mathcal{D}(X)$, and $\lambda > 0$.
(R4) The map ϕ is *equivariant*, i.e. $\phi(d^\tau) = (\phi \circ d)^\tau$ for all τ in the permutation group of X and $d \in \mathcal{D}(X)$, where $d^\tau(x, y) = d(\tau(x), \tau(y))$ for all $x, y \in X$.
(R5) If $d \in \mathcal{D}(X)$, then $\phi(d)$ can be computed in time that is polynomial in $|X|$.

Requirements (R1)–(R5) are chosen since they are desirable in biological applications: for example, (R4) can be rephrased as requiring that the tree construction method does not depend on the order in which the taxa set X is processed—a property that does not hold for the popular Neighbor Joining method, for example. (See [5–7] for more details.)

In [8], Buneman gives a method for tree construction that satisfies (R1)–(R5). However, "the price paid for continuity," as Buneman puts it, is that the resulting tree is often highly unresolved. In [5] the Buneman construction is modified in an attempt to address this problem. The resulting construction is called the *refined Buneman tree* and is shown to satisfy (R1)–(R4). However it is not shown whether (R5) holds for this construction or not [1]. In this note we fill in this gap and present an algorithm for computing the refined Buneman tree in polynomial time.

2 Buneman Trees

Given an X-tree $T = (V, E, L)$, an edge $e \in E$ induces a *split* of X (that is, a bipartition of X into two non-empty subsets) in a natural way: we say that x, y are in the same element of a split if the unique path in T from $L(x)$ to $L(y)$ does not traverse e. Clearly, the splits associated to an X-tree are *pairwise compatible*: for each pair of splits $\{U, V\}, \{U', V'\}$ at least one of the intersections $U \cap U'$, $U \cap V'$, $V \cap U'$, $V \cap V'$ is empty. We call a set of splits *compatible* if they are pairwise compatible. One can always associate a unique X-tree to a compatible set of splits of X [8]. This X-tree can be constructed in time linear in $|X|$ and the number of splits [10]. From here on we will not necessarily differentiate between a compatible set of splits and the unique

[1] Note that the algorithm currently used for computing the refined Buneman tree in the phylogenetic analysis program SplitsTree [9] has exponential time complexity.

X-tree associated to it.

In [8] Buneman actually presents a method for associating a tree-metric to a distance d on X: Define the *Buneman score* of a quartet $q = ab|cd$, $a, b, c, d \in X$ to be

$$\beta_q = \beta_{ab|cd} := \tfrac{1}{2}(\min\{ac + bd, ad + bc\} - (ab + cd)),$$

where $xy := d(x, y)$ for $x, y \in X$, and the *Buneman index* of a split $\sigma = \{U, V\}$ of X to be

$$\mu_\sigma = \mu_\sigma(d) = \min_{u,u' \in U, v,v' \in V} \beta_{uu'|vv'}.$$

Here u and u' need not be distinct; likewise for v and v'. Buneman shows that the set of splits

$$B(d) := \{\sigma \mid \mu_\sigma(d) > 0\}$$

is compatible. We define the *Buneman tree* to be the weighted X-tree associated to $B(d)$, with the edge corresponding to the split σ weighted by $\mu_\sigma(d)$ for all $\sigma \in B(d)$. The map which associates the Buneman tree to a distance function satisfies (R1)–(R5) given in the introduction.

For a general distance d the cardinality of $B(d)$ tends to be small in which case the Buneman tree is highly unresolved. It was shown in [5] that a special relaxation of the condition $\mu_\sigma > 0$ also gives a set of compatible splits: Put $n := |X|$ and let $\sigma = \{U, V\}$ be a split of X. We assume $n \geq 4$. If σ is non-trivial, that is $|U|, |V| > 1$, then define

$$Q(\sigma) := \{uu'|vv' : u, u' \in U, u \neq u', v, v' \in V, v \neq v'\}.$$

If σ is trivial then, without loss of generality, we have $|U| = 1$ and we define

$$Q(\sigma) := \{uu|vv' : u \in U, v, v' \in V, v \neq v'\}.$$

Let $q_1, \ldots, q_{|Q(\sigma)|}$ be an ordering of the elements in $Q(\sigma)$ such that for all $1 \leq i \leq j \leq |Q(\sigma)|$ we have $\beta_{q_i} \leq \beta_{q_j}$. The *refined Buneman index* of σ is defined as

$$\overline{\mu}_\sigma = \overline{\mu}_\sigma(d) := \frac{1}{n-3} \cdot \sum_{i=1}^{n-3} \beta_{q_i}.$$

The set of splits

$$RB(d) := \{\sigma \,|\, \overline{\mu}_\sigma(d) > 0\}$$

is shown to be compatible in [5] and the associated weighted X-tree, with the edge corresponding to the split σ assigned weight $\overline{\mu}_\sigma(d)$ for all $\sigma \in RB(d)$, is called the *refined Buneman tree*. It is clear that $B(d) \subseteq RB(d)$, and often $B(d)$ is strictly contained in $RB(d)$, in which case the refined Buneman tree refines the Buneman tree. The map which associates the refined Buneman tree to a distance function satisfies (R1)–(R4). To show that (R5) also holds, we first need to introduce another variation of the Buneman tree.

3 Anchored Buneman Trees

Fix $x \in X$. Given a split $\sigma = \{U, V\}$ with $x \in U$ define

$$\mu_\sigma^x = \mu_\sigma^x(d) := \min_{u \in U, v, v' \in V} \{\beta_{xu|vv'}\},$$

and put $B_x(d) := \{\sigma : \mu_\sigma^x > 0\}$. Clearly $\mu_\sigma^x \geq \mu_\sigma$ for all splits σ, so that $B(d) \subseteq B_x(d)$.

Lemma 1 *The set of splits $B_x(d)$ is compatible.*

PROOF. Choose any two non-trivial splits $\sigma = \{U, V\}$ and $\hat{\sigma} = \{U', V'\}$ in $B_x(d)$ such that $x \in U, U'$, and suppose that σ and $\hat{\sigma}$ are not compatible. Then there exist $w, y, z \in X$ such that $w \in U \cap V'$, $y \in V \cap U'$, and $z \in V \cap V'$. The quartet $xw|yz$ is in $Q(\sigma)$ and so by the definition of $B_x(d)$ we have $\beta_{xw|yz} > 0$. But we also have $xy|wz \in Q(\hat{\sigma})$ so $\beta_{xy|wz} > 0$, a contradiction. Hence σ and $\hat{\sigma}$ are compatible, from which it follows that $B_x(d)$ is compatible. □

We call the weighted X-tree associated to $B_x(d)$, with the edge corresponding to the split σ weighted by $\mu_\sigma^x(d)$ for all $\sigma \in B_x(d)$, the *Buneman tree anchored at x*. The following iterative procedure, based on algorithms in [11] and [12], constructs an anchored Buneman tree in $O(n^4)$ time. Let d be a distance on the ordered set $X = \{x = x_0, x_1, x_2, \ldots, x_n\}$.

Algorithm AnchoredBuneman(X,x,d)
1. If $d_{xx_1} > 0$ then put $\mathcal{S}_1 := \{\{\{x\}, \{x_1\}\}\}$ else put $\mathcal{S}_1 := \emptyset$.
2. For k from 2 to n do
3. Put $\mathcal{S}_k := \emptyset$.

For each split $\{U, V\} \in \mathcal{S}_{k-1}$ with $x \in U$ do
If $\beta_{xx_k|vv'} > 0$ for all $v, v' \in V$ then add $\{U \cup \{x_k\}, V\}$ to $\mathcal{S}_k$.
If $\beta_{xu|x_k v} > 0$ for all $u \in U$ and $v \in V$ then
add $\{U, V \cup \{x_k\}\}$ to $\mathcal{S}_k$.
If $\beta_{xx_j|x_k x_k} > 0$ for all $j = 1, \ldots, k-1$ then
add $\{\{x_k\}, \{x, x_1, \ldots, x_{k-1}\}\}$ to $\mathcal{S}_k$.
end (For k from 2 to n).
Output $B_x(d) = \mathcal{S}_n$ and $\{\mu^x_\sigma : \sigma \in \mathcal{S}_n\}$.
end.

The Buneman tree anchored at x satisfies (R1), (R2), (R3) and (R5). It clearly does not satisfy (R4) because it depends on the choice of taxon x. One might think that a possible way to avoid this problem would be to take the strict consensus of the anchored Buneman trees for every $x \in X$ (cf. [13]). However, we now see that this brings us straight back to the Buneman tree.

Proposition 2 *If d is a distance function on X, then*

$$B(d) = \bigcap_{x \in X} B_x(d).$$

PROOF. For all x we have $B(d) \subseteq B_x(d)$, so that $B(d) \subseteq \cap_{x \in X} B_x(d)$. To see the reverse inclusion, note that if $\{U, V\} \notin B(d)$ then there is some quartet $uu'|vv'$ such that $u, u' \in U$, $v, v' \in V$ and $\beta_{uu'|vv'} \leq 0$. It follows that $\{U, V\} \notin B_u(d)$, and so $\{U, V\} \notin \cap_{x \in X} B_x(d)$, which completes the proof. □

4 A Polynomial Algorithm for Constructing $RB(d)$

The algorithm AnchoredBuneman utilizes a useful property of the Buneman tree anchored at x: if $\{U, V\}$ is a split in $B_x(d)$, $y \in X - \{x\}$ and $y \in U$, then $\{U - \{y\}, V\}$ is in $B_x(d|_{X - \{y\}})$, where $d|_{X-\{y\}}$ is the distance d restricted to $(X - \{y\})$. The same property does *not* hold for the refined Buneman tree. We therefore require a different reduction step. First note that when $|X| = 4$, and d is a distance on X then, by definition, $B(d) = RB(d)$.

Proposition 3 *Suppose that $|X| > 4$, and fix $x \in X$. If $\sigma = \{U, V\}$ is a split in $RB(d)$ with $x \in U$, and $|U| > 2$, then either $\{U, V\} \in B_x(d)$ or $\{U - \{x\}, V\} \in RB(d|_{X-\{x\}})$ or both.*

PROOF. Suppose that $|U| > 2$ and that σ is not contained in $B_x(d)$, that is, there exists a quartet $xu|vv'$ in $Q(\sigma)$ such that $\beta_{xu|vv'} \leq 0$. Put $\hat{\sigma} = \{U - \{x\}, V\}$, so that $\hat{\sigma}$ is a split of $X - \{x\}$. We claim that $\overline{\mu}_\sigma < \overline{\mu}_{\hat{\sigma}}$.

Let $q_1, q_2, \ldots, q_{|Q(\hat{\sigma})|}$ be an ordering of $Q(\hat{\sigma})$ such that $1 \leq i \leq j \leq |Q(\hat{\sigma})|$ implies that $\beta_{q_i} \leq \beta_{q_j}$. Since $x \notin X - \{x\}$ we have $xu|vv' \notin Q(\hat{\sigma})$. Let Q^* be the set of $(n-3)$ quartets $\{q_1, q_2, \ldots, q_{n-4}\} \cup \{xu|vv'\}$. Then, by the definition of $\overline{\mu}_{\hat{\sigma}}$, we have

$$\begin{aligned}(n-4)\,\overline{\mu}_{\hat{\sigma}} &= \sum_{q \in Q^* - \{xu|vv'\}} \beta_q \\ &= \sum_{q \in Q^*} \beta_q - \beta_{xu|vv'} \\ &\geq \sum_{q \in Q^*} \beta_q,\end{aligned}$$

where the last inequality holds because $\beta_{xu|vv'} \leq 0$.

As $|U| > 2$, we clearly have $Q(\hat{\sigma}) \subseteq Q(\sigma)$ and $Q^* \subseteq Q(\sigma)$. Therefore

$$\begin{aligned}\sum_{q \in Q^*} \beta_q &\geq \min_{Z \subseteq Q(\sigma), |Z| = n-3} \left(\sum_{q \in Z} \beta_q \right) \\ &= (n-3)\,\overline{\mu}_{\sigma},\end{aligned}$$

from which it follows that $\overline{\mu}_{\hat{\sigma}} > \overline{\mu}_{\sigma}$, thus proving the claim.

Since $\sigma \in RB(d)$ we have $\overline{\mu}_{\sigma} > 0$ and since $\overline{\mu}_{\hat{\sigma}} > \overline{\mu}_{\sigma}$ we must also have $\overline{\mu}_{\hat{\sigma}} > 0$ and therefore $\hat{\sigma} \in RB(d|_{X-\{x\}})$. □

We now present an iterative algorithm for computing the set of splits $RB(d)$ for a distance d, based on the reduction step obtained in the last proposition. Assume that $|X| > 4$, and order $X = \{x_1, \ldots, x_n\}$. Put $X_k = \{x_1, \ldots, x_k\}$, $k = 1, \ldots, n$, and let d_k denote d restricted to X_k.

Algorithm RefinedBuneman(X,d)

1. Construct the list Q_X of all possible quartets on X, sorted according to their Buneman score $\beta_{ab|cd}$.
2. Let $\mathcal{S}_4 := B(d_4)$
3. For k from 5 to n do
4. Let

$$\mathcal{S}_k := \{\{\{x_i, x_k\}, X_k - \{x_i, x_k\}\} \,:\, i = 1, \ldots, k-1\} \cup \{\{x_k\}, X_k - \{x_k\}\}.$$

5. For every split $\{U, V\}$ in $\mathcal{S}_{k-1}$ do
6. Add the splits $\{U \cup \{x_k\}, V\}$ and $\{U, V \cup \{x_k\}\}$ to $\mathcal{S}_k$.
7. end(For every split)
8. Construct $B_{x_k}(d_k)$ and add in all of these splits to $\mathcal{S}_k$.

9. Remove from $\mathcal{S}_k$ all those splits $\sigma \in \mathcal{S}_k$ with $\overline{\mu}_\sigma \leq 0$.
10. end (For k).
11. Output $RB(d) = \mathcal{S}_n$ and $\{\overline{\mu}_\sigma : \sigma \in \mathcal{S}_n\}$.
end.

The correctness of this algorithm follows from Proposition 3: the splits $\{U, V\} \in RB(d_k)$ with $x_k \in U$ and $|U| \leq 2$ are included in Step 4. We now show that this algorithm takes polynomial time in $n := |X|$.

Theorem 4 *If d is a distance on X and $n \geq 4$, then the algorithm* Refined-Buneman *constructs the refined Buneman tree in at most $O(n^6)$ time.*

PROOF. Step 1 takes $O(n^4 \log n)$ time, and Step 2 takes only constant time. We now consider the steps within the loop consisting of Steps 3 to 10, for $4 < k \leq n$.

In Step 4, $\mathcal{S}_k$ is initialized to contain exactly k splits, taking $O(k)$ time. Steps 5 to 7 add two splits to $\mathcal{S}_k$ for every split in $\mathcal{S}_{k-1}$. Since $\mathcal{S}_{k-1}$ is compatible, this is at most $O(k)$ extra splits. In Step 8 we use the algorithm Anchored-Buneman to construct $B_{x_k}(d_k)$ in $O(k^4)$ time. Since $B_{x_k}(d_k)$ is compatible it contains at most $O(k)$ splits and Step 8 adds at most $O(k)$ splits to $\mathcal{S}_k$. Thus, after Steps 4 to 8, $\mathcal{S}_k$ contains at most $O(k)$ splits.

In Step 9 we have to calculate the refined Buneman indices of the $O(k)$ splits in $\mathcal{S}_k$. This we do in $O(kn^4)$ time as follows: For each split $\sigma \in \mathcal{S}_k$ proceed in ascending order through the list Q_X until $n-3$ of the quartets in $Q(\sigma)$ have been encountered. Use these $n-3$ quartets to calculate the refined Buneman index for σ. As there are $O(n^4)$ quartets in Q_X, this takes $O(n^4)$ time for each split, and since there are $O(k)$ splits, we require $O(kn^4)$ time.

Collecting these facts together, we see that each iteration of the loop consisting of Steps 3 to 10 takes $O(k + 1 + k^4 + kn^4) = O(kn^4)$ time. Thus, since we iterate this loop $n-4$ times, the algorithm takes at most $O(n^6)$ time. $\square$

Acknowledgement

The authors wish to thank the FSPM-Strukturbildungsprozesse for its generous support during this project. We also thank the referee for a quick and helpful review.

References

[1] J. Barthélemy, From copair hypergraphs to median graphs with latent vertices, *Discrete Math.* **76** (1989) 9–28.

[2] A. Dress, V. Moulton and M. Steel, Trees, taxonomy and strongly compatible multi-state characters, *Advances in Applied Mathematics* **19** (1997) 1–30.

[3] M. Nei, Relative efficiencies of different tree-making methods for molecular data, in: M. Miyamoto and J. Cracraft, eds., *Phylogenetic Analysis of DNA Sequences* (Oxford University Press, 1991) 90–128.

[4] D. Swofford, G. Olsen, P. Waddell and D. Hillis, Phylogenetic Inference, in: D. Hillis, C. Moritz and B. Mable, eds., *Molecular Systematics* (Sinauer, Sunderland, Mass, 1996) 407–514.

[5] V. Moulton and M. Steel, Retractions of finite distance functions onto tree metrics, submitted to: *Discrete Applied Mathematics* (1997).

[6] V. Moulton, M. Steel, C. Tuffley, Dissimilarity maps and substitution models, to appear in: *Proceedings of the DIMACS Workshop on Mathematical Hierarchies* (1997).

[7] K. Wolf and P.O. Degens, On properties of additive tree algorithms, in: O. Opitz, ed., *Conceptual and Numerical Analysis of Data* (Springer-Verlag, 1989) 256–265.

[8] P. Buneman, The recovery of trees from measures of dissimilarity, in: F. Hodson, D. Kendall, and P. Tautu, eds., *Mathematics in the Archaeological and Historical Sciences* (Edinburgh University Press, Edinburgh, 1971) 387–395.

[9] D. Huson, SplitsTree - a program for analyzing and visualizing evolutionary data, to appear in: *CABIOS* (1997).

[10] D. Gusfield, Efficient algorithms for inferring evolutionary trees, *Networks* **21** (1991) 19–28.

[11] H.-J. Bandelt and A. Dress, Weak hierarchies associated with similarity measures— an additive clustering technique, *Bulletin of Mathematical Biology* **51(1)** (1989) 133–166.

[12] V. Berry and O. Gascuel, Inferring evolutionary trees with strong combinatorial evidence, in: T. Jiang, ed. *Proceedings of the third international Computing and Combinatorics Conference (COCOON)* (Shanga, 1997) 111–123.

[13] F. R. McMorris, D. B. Meronk and D. A. Neumann, A view of some consensus methods for trees, in: J. Felsenstein, ed., *Numerical Taxonomy* (Springer-Verlag, 1983) 122–125.