JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 20, Number 1, 2013

Pp. 30–37
DOI: 10.1089/cmb.2012.0219

Live Phylogeny

GUILHERME P. TELLES,[1] NALVO F. ALMEIDA,[2] ROSANE MINGHIM,[3]
and MARIA EMILIA M.T. WALTER[4]

ABSTRACT

The live phylogeny problem generalizes the phylogeny problem while admitting the existence of living ancestors among the taxonomic objects. This problem suits the case of fast-evolving species, like virus, and the construction of phylogenies for nonbiological objects like documents, images, and database records. In this article, we formalize the live phylogeny problem for distances and character states and introduce polynomial-time algorithms for particular versions of the problems. We believe that more general versions of the problems are NP-hard and that many heuristic and approximation approaches may be developed as solution strategies.

Key words: algorithms, character states phylogeny, distance-based phylogeny, phylogenetic trees.

1. INTRODUCTION

Phylogeny is a core problem in computational molecular biology. Starting with a set of taxonomic objects, the problem is to reconstruct their evolutionary history. The result is a tree in which taxonomic objects are leaves and hypothetical ancestors are added as internal nodes (Felsenstein, 2004; Gusfield, 1997; Setubal and Meidanis, 1997).

This article introduces the problem of live phylogeny, where a phylogenetic tree must be reconstructed but ancestors are present among the input taxonomic objects. This way, internal nodes in the resulting tree may be either actual objects or hypothetical ancestors. Real-world applications are the analysis of viral populations or other fast-evolving organisms (Castro-Nallar et al., 2012; Gojobori et al., 1990), and the phylogenetic analysis of nonbiological objects, such as documents, images, or relational database entries (Cuadros et al., 2007; Paiva et al., 2011). We present the problem both for distances and characters. For distances, we investigated the case in which the matrices are additive. For characters, we considered absence of convergence and reversals. We give polynomial algorithms for both problems. To our best knowledge, this is the first characterization of these problems in phylogeny.

This article is organized as follows. Section 2 is devoted to the distance-based live phylogen problem, and Section 3 to the character states live phylogeny problem. In Section 4, we present some conclusions.

[1]Institute of Computing, University of Campinas, Campinas Brazil.
[2]School of Computing, Federal University of Mato Grosso do Sul, Campo Grande, Brazil.
[3]Institute of Mathematical and Computer Science, University of São Paulo, São Carlos, Brazil.
[4]Department of Computer Science, University of Brasília, Brasília, Brazil.

2. DISTANCE-BASED LIVE PHYLOGENY

In the distance-based phylogeny problem, one wants to build an unrooted, weighted tree in which the distances among leaves are equal to the distances given in a distance matrix. The input is an $n \times n$ matrix M, where $M_{i,j}$ is the distance between objects o_i and o_j. The output is a tree in which each leaf represents an object and all internal nodes have degree 3. When it is possible to build such a tree, then the distances in M are said to be additive.

It is known that if M is additive, then a polynomial algorithm solves the problem (Setubal and Meidanis, 1997). It is also known that a distance matrix M is additive if it is a metric space and respects the *four-point condition*, which states that given any four objects, it is possible to label them as o_i, o_j, o_k, o_l, such that

$$M_{i,j} + M_{k,l} = M_{i,k} + M_{j,l} \geq M_{i,l} + M_{j,k}.$$

By the other side, minimizing the nonadditivity deviation is an NP-hard problem (Day, 1987).

In distance-based live phylogeny, objects may be represented by internal nodes of the tree as well, in order to reflect living ancestors in the set of objects.

Formally, let M^n be a square matrix of order n, representing objects $o_1 \ldots o_n$, where $M^n_{i,j} \in \mathbb{R}$ is the distance between objects o_i and o_j. Let T^n be a weighted, unrooted tree. T^n is a live phylogeny for M^n if T^n is compatible with M^n. A tree T^n is compatible with a matrix M^n, denoted $T^n \sim M^n$, if

- each leaf of T^n is labeled with one object,
- each object labels exactly one node, and
- $d^n_{o_i o_j} = M^n_{ij}$, $1 \leq i, j \leq n$, where d^n_{xy} is the distance between x and y in T^n, given by the sum of the lengths of the edges in the path between x and y in T^n.

Internal nodes are called *ancestors*. An ancestor is *live* if it is labeled o_i, for some i, and is *hypothetical* otherwise.

The distance-based live phylogeny problem is, given M^n additive, to build a live phylogeny T^n. Here we provide a constructive proof that a live phylogeny can always be built from an additive matrix M^n.

Theorem 1. *Let M^k additive and T^k be such that $M^k \sim T^k$. Let M^{k+1} additive be M^k with a new object o_{k+1} distinct from every o_i, $1 \leq i \leq k$, added to it. We can add o_{k+1} to T^k, obtaining $T^{k+1} \sim M^{k+1}$.*

Proof: By induction on $k \geq 2$. ■

Basis

Suppose $k = 2$ and let x, y be the only two leaves of T^2. Let $z = o_3$ be the new node to be added to T^2, obtaining T^3. We have the four following possible cases, based on the relationships among x, y, z in T^3.

Case 1: $M^3_{xy} = M^3_{xz} + M^3_{zy}$. In this case, add z to the edge (x, y) of T^2 in such way that $d^3_{xz} = M^3_{xz}$ and $d^3_{zy} = M^3_{zy}$, obtaining T^3 (Fig. 1).

Case 2: $M^3_{xz} = M^3_{xy} + M^3_{yz}$. In this case, add a new edge (y, z) to T^2 such that $d^3_{yz} = M^3_{yz}$ and $d^3_{xz} = M^3_{xz}$, obtaining T^3 (Fig. 2).

Case 3: $M^3_{yz} = M^3_{xy} + M^3_{xz}$. In this case, add a new edge (z, x) to T^2 such that $d^3_{zx} = M^3_{zx}$ and $d^3_{yz} = M^3_{zy}$, obtaining T^3 (Fig. 3).

Notice that Cases 1, 2, and 3 are exclusive, otherwise $z = x$ or $z = y$, and we are assuming that all objects are distinct.

Case 4: When none of the previous cases happens, and because of triangle inequality, we have

$$d^3_{xz} + d^3_{zy} > d^3_{xy}, d^3_{xy} + d^3_{yz} > d^3_{xz}, d^3_{xy} + d^3_{xz} > d^3_{yz}.$$

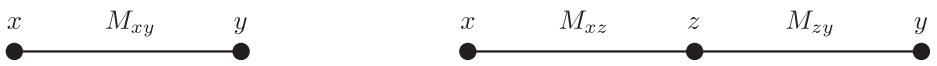

FIG. 1. In Case 1, node z is a live ancestor of x and y.

FIG. 2. In Case 2, node y became a live ancestor of x and z.

FIG. 3. In Case 3, node x becomes a live ancestor of y and z.

In this case, add a new internal node c on the edge (x, y) and connect it to z obtaining T^3, such that

$$d^3_{xc} = \frac{M^3_{xy} + M^3_{xz} - M^3_{yz}}{2} > 0,\ d^3_{yc} = \frac{M^3_{xy} + M^3_{yz} - M^3_{xz}}{2} > 0,\ d^3_{zc} = \frac{M^3_{xz} + M^3_{yz} - M^3_{xy}}{2} > 0$$

(Fig. 4). This completes the basis.

Inductive step

Suppose that $T^k \sim M^k$, $k \geq 3$. We will show how to add a new node z to T^k, obtaining $T^{k+1} \sim M^{k+1}$. Let x, y be any two leaves of T^k. Again, we have four possibilities.

In favor of a clearer notation, we denote M^{k+1} by M, d^{k+1} by d, and the only path connecting any nodes x and y in a tree by (x, y)-path.

Case i: $M_{xz} + M_{zy} = M_{xy}$. In this case, z must be inserted in the (x, y)-path, such that $d_{xz} = M_{xz}$ and $d_{zy} = M_{zy}$. It is exactly the same situation shown in Figure 1, except that now we are handling an (x, y)-path, not necessarily an edge (x, y).

Let us suppose there is no node in the position where z has been added. The case in which there is already such a node will be seen soon.

We need to show that $d_{zw} = M_{zw}$ for any node $w \neq x, y$. Suppose that w is not in (x, y)-path, but it is connected to it by a node c in the (z, y)-path (Fig. 5). The case where c is in (x, z)-path is analogous.

From the tree $d_{wx} + d_{wy} - d_{xy} = 2d_{cw} = d_{wz} + d_{wy} - d_{zy}$, so $d_{wz} = d_{wx} - d_{xy} + d_{zy}$.

The four-point condition for these points can be verified by the labeling that results in $M_{xw} + M_{zy} = M_{xy} + M_{zw}$. Thus, $M_{wx} - M_{xy} = M_{zw} - M_{zy}$. Then

$$\begin{aligned} d_{zw} &= d_{wx} - d_{xy} + d_{zy} \\ &= M_{wx} - M_{xy} + d_{zy} \quad \text{(induction hypothesis)} \\ &= M_{zw} - M_{zy} + d_{zy} \quad \text{(rewriting } M_{wx} - M_{xy}\text{)} \\ &= M_{zw} - M_{zy} + M_{zy} \quad \text{(construction)} \\ &= M_{zw} \end{aligned}$$

Note that this proof works also for the case in which $c = w$.

Finally, let us see what happens when there is already an internal node c in the position where z should be added. Node c is a hypothetical ancestor, otherwise z would be already in the tree. It is enough to transform this internal node c into the live ancestral $z = o_{k+1}$. We only need to show that $d_{zw} = M_{zw}$, for any live node w connected to z without using edges or nodes in (x, y)-path, because the situations in which there are nodes from (x, y)-path are covered above. (Fig. 6).

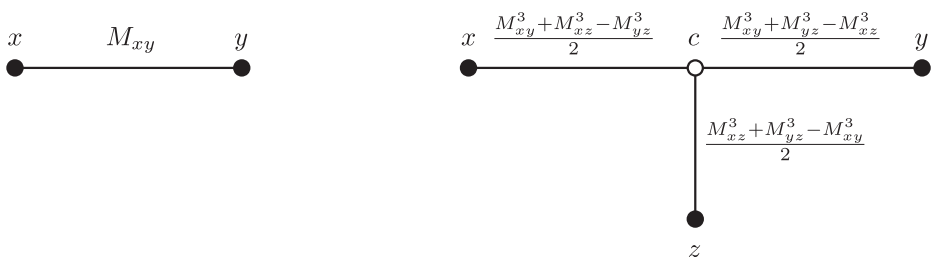

FIG. 4. In Case 4, there is a hypothetical ancestor c of x, y, and z.

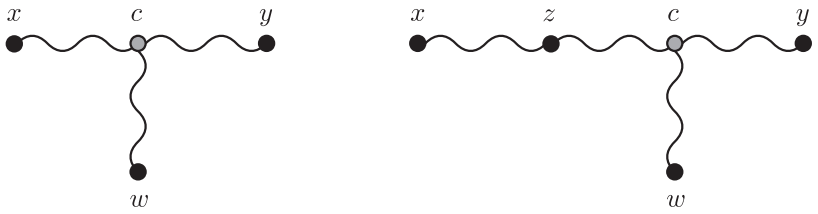

FIG. 5. Case i when w is not in (x, y)-path but it is connected to it by a node c and the new z is in (c, y)-path.

The four-point condition for these points can be verified by the labeling that results in $M_{zw} + M_{yx} = M_{xz} + M_{wy}$. Then,

$$\begin{aligned}
2d_{zw} &= d_{xw} + d_{yw} - d_{xy} && \text{(from } T^{k+1}\text{)}\\
2d_{zw} &= d_{xw} + d_{yw} - M_{xy} && \text{(induction hypothesis)}\\
2d_{zw} &= d_{xw} + d_{yw} - M_{xz} - M_{wy} + M_{zw} && \text{(rewriting } M_{xy}\text{)}\\
2d_{zw} &= d_{xw} + M_{yw} - M_{xz} - M_{wy} + M_{zw} && \text{(induction hypothesis)}\\
2d_{zw} &= d_{xw} - d_{xz} + M_{zw} && \text{(construction)}\\
2d_{zw} &= d_{zw} + M_{zw} && \text{(from } T^{k+1}\text{)}\\
d_{zw} &= M_{zw}
\end{aligned}$$

This concludes Case i.

Case ii: $M_{xz} = M_{xy} + M_{yz}$. This case is similar to Case 2 of the basis, in the sense that z is added to T by connecting it to y through a new edge (y, z).

We need to show that, $d_{zw} = M_{zw}$, for any node w in T^{k+1}, $w \neq x, y$. Let $w \neq x, y$ be a node of T^{k+1} and c the node connecting w to the path that connects x to y. (Fig. 7).

The four-point condition for these points can be verified by the labeling that results in $M_{xy} + M_{wz} = M_{xz} + M_{yw}$. Then,

$$\begin{aligned}
d_{zw} &= d_{wy} + d_{yz} && \text{(from the tree)}\\
&= M_{wy} + d_{yz} && \text{(induction hypothesis)}\\
&= M_{xy} + M_{wz} - M_{xz} - d_{yz} && \text{(rewriting } M_{yw}\text{)}\\
&= d_{xy} + M_{wz} - d_{xz} + d_{yz} && \text{(induction hypothesis)}\\
&= M_{wz}
\end{aligned}$$

Notice that this proof holds also in the case where $w = c$.

Case iii: $M_{yz} = M_{xy} + M_{xz}$. This case is similar to Case 3 of the basis, and node z is added on the same way. The proof is analogous to the one given in Case ii.

If none of the cases i, ii, and iii happens, then we try to add the new node z to T^k through an edge connecting z and a node c in the (x, y)-path, as we do in Case 4 of the basis. There are three possibilities to consider.

Case iv-a: There is no node c in T^k as it also happens in Case 4 of the basis. We create this new node c as a hypothetical ancestor and connect c to z through a new edge (c, z) with length $(M(z, x) + M(z, y) - M(x, y))/2$.

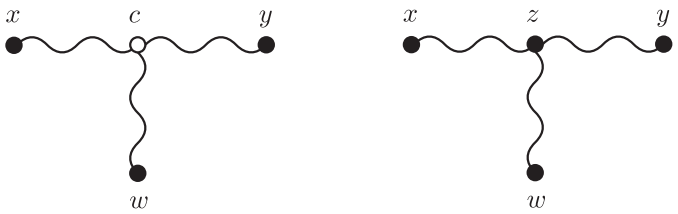

FIG. 6. Case i when a hypothetical c exists at z's position, and c is replaced by z.

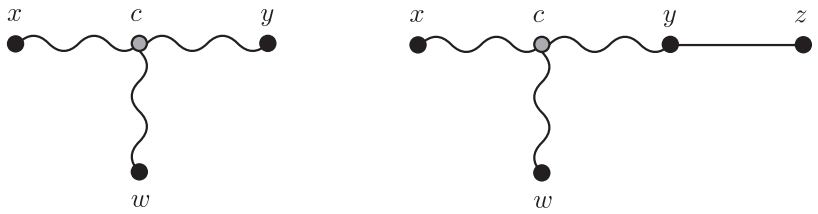

FIG. 7. In Case ii, the new node z is connected to y by a new edge.

We need to show that $d_{wz} = M_{wz}$ for every $w \neq x, y$. Let us suppose that there is a path from v to w, where v is a node in the (x, c)-path. The case where v is in the (c, y)-path is analogous (Fig. 8).

The four-point condition for these points can be verified by the labeling that results in $M_{xy} + M_{wz} = M_{xz} + M_{yw}$. From the tree, $d_{zw} + d_{zy} - d_{wy} = 2d_{cz} = d_{xz} + d_{zy} - d_{xy}$. So $d_{zw} = d_{xz} - d_{xy} + d_{wy}$.

$$\begin{aligned} d_{zw} &= d_{xz} - d_{xy} + d_{wy} \\ &= d_{xz} - M_{xy} + M_{wy} && \text{(induction)} \\ &= d_{xz} + M_{wz} - M_{xz} && \text{(rewriting } M_{wy} - M_{xy}\text{)} \\ &= M_{xz} + M_{wz} - M_{xz} && \text{(construction)} \\ &= M_{wz} \end{aligned}$$

Note that this proof works also for the case in which $v = w$.

Case iv-b: Suppose that there is already a node c in T^k, as in Case 4 of the basis, such that c has degree 2. Because we only create a hypothetical node with degree 3, c is a live ancestor. In this case, we just add z to T^k and connect z to c through a new edge (z, c) with the same length as in Case iv-a. The proof that $d_{wz} = M_{wz}$ for every $w \neq x, y$ is similar to that provided for Case iv-a.

Case iv-c: Now, consider the case in which there is a node c in T^k, as in Case 4 of the basis, but c has degree >2. This means that there is at least another leaf w connected to c through a path not using any vertex or edge in (x, y)-path. To solve this case, find any pair of leaves r, s such that c is in (r, s)-path and one of the previous cases i, ii, iii, iv-a, and iv-b holds, and apply the appropriated case. If there is no such pair of leaves r, s, then just add z to the tree and connect it to c through a new edge (c, z).

We need to show that $d_{wz} = M_{wz}$ for every $w \neq x, y$. If w is either in the (x, y)-path, or there is a path from d to w, where d is a node in the (x, c)-path, then the proofs are similar to the previous ones. Otherwise w is in a path connected to c, as shown in Figure 9.

The four-point condition for these points can be verified by the labeling that results in $M_{xy} + M_{wz} = M_{xz} + M_{yw}$. From the tree, $d_{xz} + d_{zy} - d_{xy} = 2d_{cz} = d_{zw} + d_{zy} - d_{wy}$, and the proof follows exactly as the one for Case iv-a. ■

The constructive proof of Theorem 1 gives us an algorithm to build the live phylogeny given an additive matrix. The algorithm consists of starting with two objects connected by an edge and applying, in each step, one of the described cases. This algorithm clearly has polynomial time in the number of objects, since the test for the correct case can be done in constant time, except for Case iv-b, where we need to find a pair of leaves satisfying any of the other cases. Because there are $O(k^2)$ possible pairs of leaves, in each step k, the total time is $O(n^3)$.

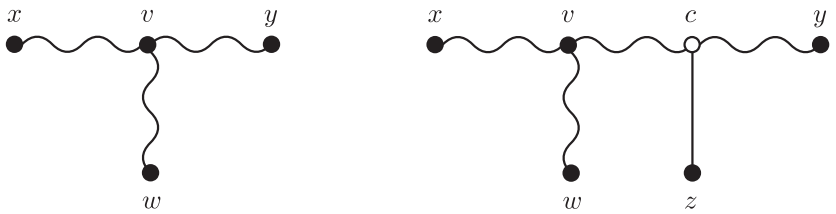

FIG. 8. In Case iv-a, the new node z is connected to a new hypothetical ancestor c by an edge.

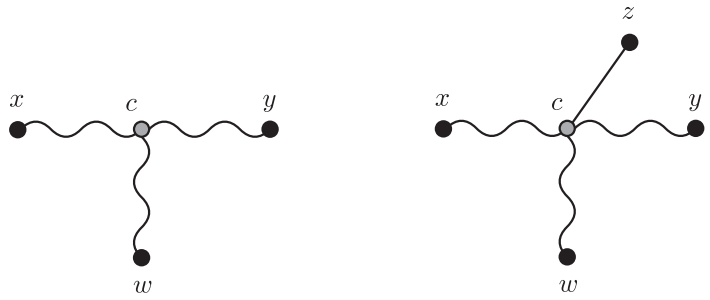

FIG. 9. In Case iv-c, the new node z is connected to an existing hypothetical ancestor c by an edge.

3. CHARACTER STATES LIVE PHYLOGENY

For this problem, one is interested in building a phylogenetic rooted tree that explains the evolutionary relationship among objects, based on states of characters that each object possesses. More formally, the input is an $n \times m$ matrix M, where $M_{i,j}$ is the state of character j for object i.

In the related literature, the character-based phylogeny problem has been approached, considering the number of possible states for each character and whether or not there is an order relation defined for character states. If the number of states is fixed, then the problem can be easily solved. Otherwise, it is NP-hard. When the order is totally defined, then there is a polynomial-time algorithm for the problem. Otherwise, the problem is also NP-hard (Setubal and Meidanis, 1997).

Another issue concerns more complicated evolutionary facts, such as reversal and parallel evolution events. A reversal happens when a character changes back to a previous state. A parallel evolution happens when a character changes to the same state in two distinct lineages. When there are no such events, the problem is known as the perfect phylogeny problem. Other works in the literature concern minimizing the number of reversals and parallel evolution events to obtain a perfect phylogeny or maximizing the number of characters that admit a perfect phylogeny (Setubal and Meidanis, 1997).

In this work, we introduce the perfect live phylogeny with two character states. Let M be an $n \times m$ binary matrix whose rows are labeled $o_1, o_2 \ldots o_n$, whose columns are labeled $c_1, c_2 \ldots c_m$, and whose columns are pairwise disjoint or comparable. A live perfect phylogeny for M is a rooted tree T such that:

1. Each edge in T is labeled with a distinct c_j;
2. each object o_i labels exactly one node in T;
3. for every node labeled o_i, $M_{i,\,j} = 1$ if and only if c_j is in the path from the root of T to the node labeled o_i.

Observe that this definition allows T to have labeled internal nodes. Then, objects given in the input may be ancestors in the phylogeny.

```
PERFECT-LIVE-PHYLOGENY(M)
 1  create root
 2  for i = 1 to n
 3       cur = root
 4       for j = 1 to m
 5            if M_ij = 1
 6                 if there is no edge (cur, u) labeled c_j
 7                      create node u and edge (cur, u) labeled c_j
 8                 cur = u
 9       label cur with o_i
10  return root
```

FIG. 10. Perfect live phylogeny polynomial-time algorithm.

	c_1	c_2	c_3	c_4	c_5	c_6
A	1	0	1	0	0	0
B	1	1	0	0	0	0
C	1	0	0	1	0	0
D	0	0	0	0	1	0
E	1	1	0	0	0	1

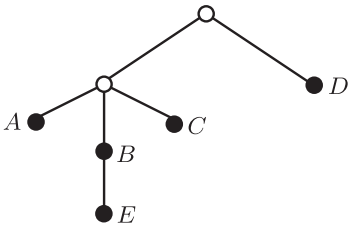

FIG. 11. A binary matrix with columns sorted in nonincreasing order of the number of 1s, and the corresponding live phylogeny tree, in which B is a live ancestor.

The polynomial time algorithm to solve this problem is presented in Figure 10. It is a simple adaptation of the algorithm for perfect phylogeny by Waterman et al. (1977). The input is a binary matrix M with columns sorted in nonincreasing order of the number of 1s. An example input and tree appear in Figure 11.

For the correctness of the algorithm, let T be the tree whose root is returned by the algorithm. It is easy to see that each o_i is used to label exactly one node of T. It is also easy to see that no edge is created without a label. Now, suppose that two edges in T were labeled c_j, while processing objects o_i and o_k, $i < k$. The ordering of the columns of M, and the fact that each pair of columns of M is either comparable or disjoint, guarantee that $M_{i,j'} = M_{k,j'}$, $1 \leq j' \leq j - 1$. Thus, by construction, the paths between the root and both edges labeled c_j must be the same, which is a contradiction.

To see that if $M_{i,j} = 1$ then c_j is in the path from the root of T to the node labeled o_i, it is enough to see that during the processing of row i of M, either the edge c_j is created or traversed. By the other side, if edge c_j is in the path between the root and the node labeled o_i, then o_i was labeled after creating or the traversing of edge c_j, and that happens only if $M_{i,j} = 1$.

4. CONCLUSIONS

Live phylogeny generalizes phylogeny while broadening its application to other areas distinct from molecular biology, such as visualization, data mining, and forensics. As with phylogeny, live phylogeny will certainly lead to NP-hard problems when the restrictions we considered here are released, namely the absence of reversals and parallel evolution, and additivity. A broad class of approximation algorithms and heuristics may then be explored for the problem. We conclude our article noting that applications beyond molecular biology deal with very large datasets, and large-scale techniques may be considered as well.

ACKNOWLEDGMENTS

G.P.T. and R.M. acknowledge the financial support of CNPq and FAPESP. N.F.A. acknowledges CNPq 305503/2010-3 grant. M.E.M.T.W. acknowledges CNPq 306731/2009-6 and FINEP 01.08.0166.00 grants.

AUTHOR DISCLOSURE STATEMENT

The authors declare that no competing financial interests exist.

REFERENCES

Castro-Nallar, E., Perez-Losada, M., Burton, G.F., et al. 2012. The evolution of HIV: Inferences using phylogenetics. *Mol. Phylogenetics Evol.* 62:777–792.

Cuadros, A.M., Paulovich, F.V., Minghim, R., et al. 2007. Point placement by phylogenetic trees and its application to visual analysis of document collections. *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology* 99–106.

Day, W.E. 1987. Computational complexity of inferring phylogenies from the similarity matrix. *Bulletin of Mathematical Biology* 49, 461–467.

Felsenstein, J. 2004. *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA.
Gojobori, T., Moriyama, E.N., and Kimura, M. 1990. Molecular clock of viral evolution, and the neutral theory. *P. Natl. Acad. Sci.* 87, 10015–10018.
Gusfield, D. 1997. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, West Nyack, NY.
Paiva, J.G.S., Florian-Cruz, L., Pedrini, H., et al. 2011. Improved similarity trees and their application to visual data classification. *IEEE T. Vis. Comput. Gr.* 17, 2459–2468.
Setubal, J.C., and Meidanis, J. 1997. *Introduction to Molecular Computational Biology*. PWS, Publishing Boston.
Waterman, M.S., Smith, T.T., Singh, M., et al. 1977. Additive evolutionary trees. *J. Theor. Biol.* 64, 199–213.

Address correspondence to:
Guilherme P. Telles
Institute of Computing
University of Campinas
Av. Albert Einstein, 1251
13083-852 Campinas-SP
Brazil

E-mail: gpt@ic.unicamp.br