JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 15, Number 8, 2008

Pp. 1–21
DOI: 10.1089/cmb.2008.0069

DUPCAR: Reconstructing Contiguous Ancestral Regions with Duplications

JIAN MA,[1] AAKROSH RATAN,[2] BRIAN J. RANEY,[1] BERNARD B. SUH,[1]
LOUXIN ZHANG,[3] WEBB MILLER,[2] and DAVID HAUSSLER[1]

ABSTRACT

Accurately reconstructing the large-scale gene order in an ancestral genome is a critical step to better understand genome evolution. In this paper, we propose a heuristic algorithm, called DUPCAR, for reconstructing ancestral genomic orders with duplications. The method starts from the order of genes in modern genomes and predicts predecessor and successor relationships in the ancestor. Then a greedy algorithm is used to reconstruct the ancestral orders by connecting genes into contiguous regions based on predicted adjacencies. Computer simulation was used to validate the algorithm. We also applied the method to reconstruct the ancestral chromosome X of placental mammals and the ancestral genomes of the ciliate *Paramecium tetraurelia*.

Key words: contiguous ancestral region, duplication, gene-order reconstruction, genome rearrangement, isometric reconciliation.

1. INTRODUCTION

THE LARGE NUMBER OF GENOME SEQUENCES becoming available makes it feasible to computationally reconstruct ancient genomes of related species that have undergone large-scale genome rearrangements. The heart of this problem is to "undo" these rearrangements and restore the ancestral order. Previous studies mainly focused on solving the median problem, which is based either on reversal (inversion) distance or breakpoint distance. In this problem one tries to reconstruct the common ancestor of two descendant genomes using an additional "outgroup" genome. Unfortunately, the median problem does not have exact and efficient algorithms (Caprara, 1999; Pe'er and Shamir, 1998). Heuristic programs for both the breakpoint median problem and the reversal median problem have been proposed (Sankoff and Blanchette, 1998; Moret et al., 2001; Bourque and Pevzner, 2002). However, the discrepancy between computational predictions and results from cytogenetic experiments (Froenicke et al., 2006; Bourque et al., 2006) suggests a need to explore further computational methods for ancestral genome reconstruction.

We recently proposed a new approach for reconstructing the ancestral order based on the adjacencies of orthologous genomic intervals in modern species (Ma et al., 2006), which essentially avoids solving any

[1]Center for Biomolecular Science and Engineering, University of California, Santa Cruz, California.
[2]Center for Comparative Genomics and Bioinformatics, Penn State University, University Park, Pennsylvania.
[3]Department of Mathematics, National University of Singapore, Singapore.

rearrangement median problem. The critical procedure of the method is analogous to Fitch's parsimony algorithm (Fitch, 1971). Instead of inferring ancestral nucleotides, we infer the locally parsimonious predecessor and successor relationships of the orthologous conserved segments in the ancestor, in our case the ancestor of most placental mammals, known as the *boreoeutherian* ancestor. Another procedure then connects these segments into 29 contiguous ancestral regions (CARs). Our result agrees with the cytogenetic prediction fairly well (Rocchi et al., 2006).

However, the main drawback of the CARs method is that it does not handle duplications. Indeed, duplications (including segmental duplications and tandem duplications) have a great impact on genome evolution (Eichler and Sankoff, 2003). Some previous theoretic studies (Sankoff, 1999; Sankoff and El-Mabrouk, 2000; Marron et al., 2004) have included duplications (sometimes with loss) along with rearrangements. In this paper, we propose an efficient heuristic approach based on the CARs method to incorporate duplication events into predictions of ancestral gene orders. Our method puts rearrangements and duplications in a unified framework in order to have a reconstruction that captures additional large-scale evolutionary events. We have applied it to reconstruct the ancestral chromosome X of placental mammals and ancestral genomes of the ciliate *Paramecium tetraurelia*.

2. METHODS

2.1. Definitions

Before applying the reconstruction algorithm, it is always important to partition the genomic region under consideration into intervals so that further rearrangement analysis can be carried out. There have been a number of approaches to identify these regions, either based on gene content or sequence similarity. Nadeau and Taylor (1984) introduced the term *conserved segment* to signify a genomic interval with gene orders that are preserved, and not disrupted by evolutionary rearrangements. In the past decade, using comparative gene mapping to find orthologous gene loci as the evolutionary markers played an important role in testing algorithms and understanding rearrangement scenarios. Recent effort has shifted to reconstructing mammalian ancestral genomes in increasingly high resolution, and whole-genome sequence alignments play a key role for identifying conserved segments (sometimes also called *synteny blocks*) (Pevzner and Tesler, 2003; Ma et al., 2006). From an evolutionary point of view, a conserved segment represents homologous segments in extant species that are derived from a common ancestral interval that was never disrupted by breakpoints caused by duplications and rearrangements.

In this paper we co-opt the word **gene** to mean a conserved segment used in the analysis. A **chromosome** of a modern or ancestral genome consists of a list of genes, where each gene has a sign (orientation) that is either positive (+) or negative (−). For a gene x, $-x$ denotes the reverse complement, and vice versa. For example, if x = `GTAT`, then $-x$ = `ATAC`. The reverse complement of a chromosome is obtained by reversing the list of genes and replacing each gene by its reverse complement. A **genome** is a set of chromosomes.

If two genes share homology, i.e., they are derived from a common ancestral gene, then they belong to the same **gene family**. We use the following notation to denote genes in genomes:

Notation. Suppose there are N gene families $a_1, a_2, \ldots, a_N$. The m-th member (according to a fixed but arbitrary ordering) of family a_i in genome g is denoted by $g[a_i^m]$.

If there is only one gene from family a_i in g, we can refer to it as $g[a_i]$. If the genome is unambiguous from the context, $g[a_i^m]$ can be simplified to a_i^m.

Definition. For a gene $g[a_i^m]$, if $g[a_j^n]$ immediately precedes $g[a_i^m]$ in the same chromosome, then we define $p(g[a_i^m]) = g[a_j^n]$ and call $g[a_j^n]$ the **predecessor** of $g[a_i^m]$. Equivalently, in the reverse complement of the chromosome, $p(g[-a_j^n]) = g[-a_i^m]$. Since $g[a_i^m]$ then immediately succeeds $g[a_j^n]$, we define $s(g[a_j^n]) = g[a_i^m]$ and call $g[a_i^m]$ the **successor** of $g[a_j^n]$. We set $p(g[a_i^m]) =$ \$ if $g[a_i^m]$ appears first on a chromosome in g, and set $s(g[a_i^m]) =$ \$ if $g[a_i^m]$ appears last on a chromosome in g. (Note that \$ is just a special symbol for which we do not define a reverse complement. Furthermore, \$ is the same for all chromosomes.)

Example. Let g have chromosome $(a\ -b^1\ -c\ b^2\ d\ e)$. Then $p(a) = \$$, $p(-b^1) = a$, $p(d) = b^2$, $s(c) = b^1$, $p(-a) = b^1$, etc.

During genome evolution, in addition to point mutations, large-scale operations can happen. These operations include insertions, deletions, chromosomal rearrangements (inversion, translocation, fusion, and fission), and duplications (tandem duplications and segmental duplications).

Suppose we have a genome $(a^1\ b^1\ c^1\ d^1, e^1\ f^1)$, where the comma separates chromosomes. Examples of the large-scale operations are as follows:

- *insertion.* $a^1\ b^1\ c^1\ d^1, e^1\ f^1 \Rightarrow a^1\ b^1\ c^1\ d^1, e^1\ \underline{h}\ f^1$.
- *deletion.* $a^1\ \underline{b^1\ c^1}\ d^1, e^1\ f^1 \Rightarrow a^1\ d^1, e^1\ f^1$.
- *inversion.* $a^1\ \underline{b^1\ c^1}\ d^1, e^1\ f^1 \Rightarrow a^1\ -c^1\ -b^1\ d^1, e^1\ f^1$.
- *translocation.* $a^1\ b^1\ \underline{c^1\ d^1}, e^1\ \underline{f^1} \Rightarrow a^1\ b^1\ f^1, e^1\ c^1\ d^1$.
- *fusion.* $a^1\ b^1\ c^1\ d^1, e^1\ f^1 \Rightarrow a^1\ b^1\ c^1\ d^1\ e^1\ f^1$.
- *fission.* $a^1\ b^1\ c^1\ d^1, e^1\ f^1 \Rightarrow a^1\ b^1, e^1\ f^1, c^1\ d^1$.
- *tandem duplication.* $a^1\ \underline{b^1\ c^1}\ d^1, e^1\ f^1 \Rightarrow a^1\ b^1\ c^1\ b^2\ c^2\ d^1, e^1\ f^1$
- *segmental duplication.* $a^1\ \underline{b^1}\ c^1\ d^1, e^1\ f^1 \Rightarrow a^1\ b^1\ c^1\ d^1, e^1\ b^2\ f^1$

As a consequence of the accumulation of the above operations, we have different numbers of genes and different gene orders in present-day genomes.

Problem. The problem we investigate in this paper is: Given (1) a set of modern genomes $\mathbb{G}$; (2) a species tree T that describes the phylogeny of these modern genomes; and (3) a set of gene trees, T_{a_i} for each gene family a_i, that defines the relationships among all the genes in the family, how can we reconstruct the order and orientation of genes in the target ancestral genome? Here, we call each reconstructed chromosome a **contiguous ancestral region (CAR)**.

2.2. The species tree and the gene trees

Species tree. A species tree is a rooted binary tree describing the phylogeny among given species. Each bifurcating ancestral node represents the genome of an ancestral species just before a speciation event, while each leaf corresponds to the genome of a modern species. Each branch in the tree, connecting genomes f and g, has a length (distance) $D(f, g)$ representing the evolutionary distance between f and g. The distance between any two nodes in the tree is the sum of the distances along the path that connects them, i.e., the distance structure on the tree is additive. The distance D is extended in the natural way to apply to points along a branch in the tree as well, which represent intermediate genomes in the descent of the species represented at the end of the branch. These intermediate genomes are the result of evolutionary operations that occur on that branch. We assume an arbitrarily long branch leading to the root of the species tree, with intermediate genomes representing ancestral forms of the root species, e.g., forms that existed prior to particular duplications that occurred in the descent of the root species from previous ancestors.

Gene tree. A gene tree is an unrooted binary tree, characterizing the relationships among genes in the same gene family across different species. Each node represents a particular gene in a gene family, from a particular genome. The branch between related genes a^n and a^n has a length (distance) $d(a^n, a^m)$, representing the evolutionary distance between a^n and a^m. Distances between arbitrary pairs of genes are defined additively, as in the species tree. If a and b are genes from different gene families, $d(a, b) = \infty$.

Distances of genes and genomes. We assume that genes and genomes evolve in the following way: When a duplication or speciation event occurs, any two homologous gene copies derived from this event begin at evolutionary distance zero. Then they independently accumulate increasing evolutionary distance as time goes by. Within any given genome, all between-gene distances increase at the same rate. However, different genomes are allowed to evolve at different rates, i.e., no universal molecular clock is assumed.

It follows from these assumptions that there is a **mapping** Φ from genes in gene trees to points in the species tree that roots each gene tree and places each gene at the position where the event that it represents (speciation or duplication) occurs. A gene from a leaf node of a gene tree is mapped to the leaf node in the species tree representing the species from which it comes. A gene from an internal node in the gene tree that represents a speciation event is mapped to the corresponding speciation node in the species tree.

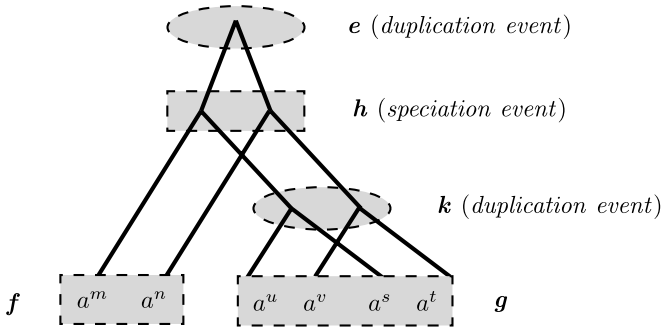

FIG. 1. The evolutionary scenario of six genes in gene family a. f and g are two present-day genomes, k and e are two duplication events, and h is a speciation event, which is also the last common ancestor of f and g. If h is the root of the species tree, then the root r of this gene tree would map to a point $\Phi(r)$ that lies above the root of the species tree, corresponding to e, on the branch leading to that root. However, if there were an outgroup species to this tree, and thus a deeper root representing an ancient genome prior to the duplication event e, then $\Phi(r)$ would lie below the root of the species tree, on the branch leading from the root to h.

A gene from an internal node in the gene tree that represents a duplication event is mapped to the point along the branch in the species tree where that duplication occurs. Finally, for each gene tree, a new node is determined on one of the branches of the gene tree that serves as the root of the gene tree, and this point is also mapped to a point in the species tree. This node in the species tree is not necessarily the root node, as the root of the gene family may trace back to a duplication that occurred before the root of the species tree (i.e., above the root of the species tree, on the branch leading to the root), or all copies of the gene may be missing in some sublineages of the species tree, putting the root of the gene tree below the root of the species tree (Fig. 1).

We can use the concept of orthology and paralogy (Fitch, 2000) to distinguish the possible relationships between two genes in the same family. If two genes diverged due to a speciation event, then they are **orthologous**. If two genes diverged due to a duplication event, then they are **paralogous**. For instance, in the example in Figure 1, $f[a^m]$ and $g[a^u]$ are orthologous, $f[a^m]$ and $g[a^s]$ are orthologous, but $f[a^m]$ and $g[a^v]$ are paralogous, and $g[a^u]$ and $g[a^s]$ are paralogous.

2.3. Isometric reconciliation

We assume that the species tree and all the gene trees are given such that branch lengths in both the species tree and the gene trees reflect the exact evolutionary distances. The species are given for each leaf of each gene tree, but not the species for internal nodes. Any mapping Φ from a gene tree B to a species tree T that roots the gene tree is an **isometric reconciliation** if

(1) Every leaf of B maps to the leaf of the designated species in T.
(2) Each internal node of B maps to a speciation node in T or a point on a branch in T.
(3) The new root r of B maps to a point $\Phi(r)$ on a branch in T such that any other node x in B maps to $\Phi(x)$ below $\Phi(r)$ and $D(\Phi(x), \Phi(r)) = d(x, r)$.

Internal nodes of the gene tree that map to speciation nodes in the species tree are determined to be speciation events by an isometric reconciliation Φ, and nodes that map to points along branches are determined to be duplication events. These determinations in turn define the relationships of orthology and paralogy for the gene family.

Theorem 1. *Given a set $T_{a_1}, \ldots, T_{a_N}$ of unrooted gene trees with designated leaf species and a species tree T, the isometric reconciliation of $T_{a_1}, \ldots, T_{a_N}$ with T is unique and there is an efficient procedure to either construct it or detect that no such isometric reconciliation exists.*

We describe our algorithm below and sketch the proof briefly. We leave the detailed proof in the Appendix.

The reconciliation algorithm we define works from the leaves of a gene tree inwards. When all the nodes in two of the three subtrees branching off from an internal node x have been processed, then x can be processed. Other than this, the order of processing is arbitrary. In the last step, all three subtrees of the last remaining unprocessed node x will already have been processed, and processing x completes the reconciliation. This reconciliation algorithm is less complicated than the traditional methods, e.g., Goodman et al. (1979) and Guigo et al. (1996). When the species tree is unknown and one wants to reconcile an optimal species tree to minimize the number of duplications using the given topologies of gene trees, the problem is **NP**-hard (Ma et al., 2000). In our case, the true species tree is known and the distances in the gene trees are exact.

Let x be an unmapped internal node of the gene tree B, with branches to nodes u, v and z. Suppose the mappings $\Phi(u)$ and $\Phi(v)$ of the nodes u and v to the species tree T have already been determined. Then the following procedure will map the node x to the species tree, or reject the input B as not reconcilable. As a side effect, if necessary the procedure will root the tree B.

MapGeneTreeNode(x). Let $d_1 = d(x, u)$, $d_2 = d(x, v)$, λ be the last common ancestor of $\Phi(u)$ and $\Phi(v)$ in T (if $\Phi(u) = \Phi(v)$ then $\lambda = \Phi(u) = \Phi(v)$), $d_1' = D(\Phi(u), \lambda)$ and $d_2' = D(\Phi(v), \lambda)$.

(1) If $d_1 + d_2 < d_1' + d_2'$, reject and exit.
(2) Let $\epsilon = d_1 + d_2 - d_1' - d_2'$ (≥ 0).
(3) If $d_1 = d_1' + \epsilon/2$ (and $d_2 = d_2' + \epsilon/2$) then map x to a point $\Phi(x)$ at distance $\epsilon/2$ above λ in the species tree T.
(4) Else if $\epsilon = 0$ then map x to a point $\Phi(x)$ at distance d_1 from $\Phi(u)$ and d_2 from $\Phi(v)$ on the path connecting $\Phi(u)$ and $\Phi(v)$ in T.
(5) Else (i.e., if $\epsilon > 0$ and $d_1 \neq d_1' + \epsilon/2$) if the gene tree B is already rooted then reject and exit
(6) Else
  (a) Place the root r of B at distance $d_1' + \epsilon/2$ from u and $d_2' + \epsilon/2$ from v on the path that connects u and v in B, and map r to a point $\Phi(r)$ at distance $\epsilon/2$ above λ.
  (b) If $d_1 < d_1' + \epsilon/2$ then map x to the point $\Phi(x)$ at distance d_1 above $\Phi(u)$, else (i.e. if $d_2 < d_2' + \epsilon/2$) map x to the point $\Phi(x)$ at distance d_2 above $\Phi(v)$.
(7) If z has already been mapped to the node $\Phi(z)$ in T, then
  (a) If the tree is already rooted, reject and exit unless $\Phi(z)$ lies below $\Phi(x)$ in T or vice-versa and $D(\Phi(x), \Phi(z)) = d(x, z)$.
  (b) Else
    i. Let $d = d(x, z)$, λ be the last common ancestor of $\Phi(x)$ and $\Phi(z)$ in T, $d_1' = D(\Phi(x), \lambda)$, $d_2' = D(\Phi(z), \lambda)$, and $\epsilon = d - d_1' - d_2'$.
    ii. If $\epsilon < 0$, reject and exit.
    iii. Else if $\epsilon = 0$ and $\Phi(x) = \lambda$, reject and exit.
    iv. Else place the root r of B at distance $d_1' + \epsilon/2$ from x and $d_2' + \epsilon/2$ from z on the path that connects x and z in B, and map r to a point $\Phi(r)$ at distance $\epsilon/2$ above λ.

We construct the reconciled tree $\mathbb{T}$ for a set of gene trees as follows (Fig. 2). We initialize $\mathbb{T}$ to be the same as species tree T. Eventually in $\mathbb{T}$, additional points will be labeled along branches based on the mapping from nodes in the gene trees.

ReconcileTrees($T, \mathsf{T}_{a_i}, i=1,2,\ldots,N$).
Input: species tree T, gene trees $\mathsf{T}_{a_1}, \mathsf{T}_{a_2}, \ldots, \mathsf{T}_{a_N}$
Output: reconciled tree $\mathbb{T}$

```
1: 𝕋 ⇐ T
2: for all gene trees T_{a_i} such that 1 ≤ i ≤ N do
3:    M ⇐ ∅ (the set of maximally internal mapped nodes)
4:    for all leaf nodes k in T_{a_i} do
5:       Φ(k) ⇐ g, where g is the genome to which k belongs.
6:       add k to M and set t_k = {k} (mapped subtree for k ∈ M).
7:    end for
```

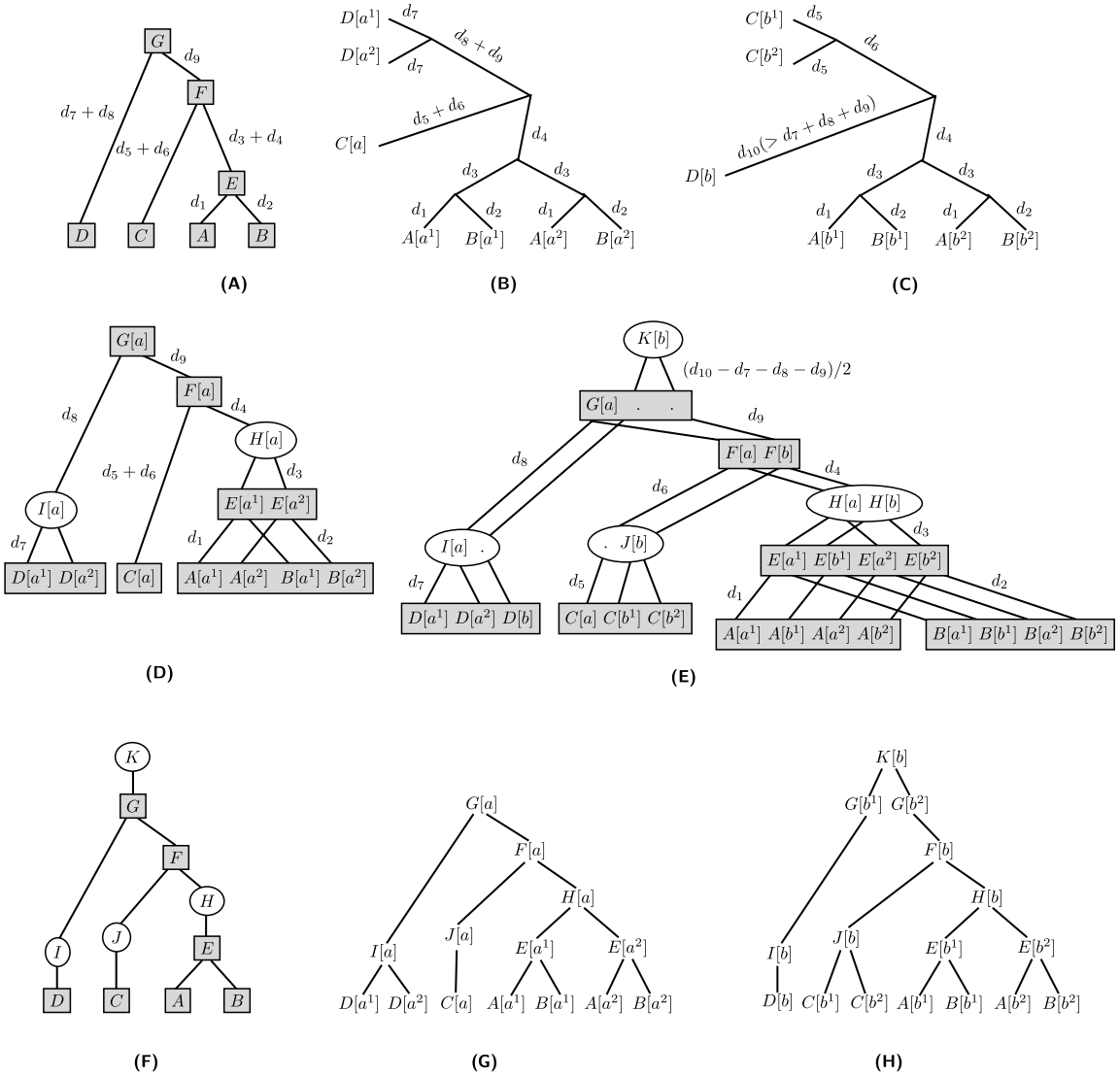

FIG. 2. **(A)** A species tree T with distances labeled to each branch. **(B)** Gene tree T_a. **(C)** Gene tree T_b. **(D)** Reconciled tree $\mathbb{T}$ after T_a being merged. **(E)** Reconciled tree $\mathbb{T}$ after both T_a and T_b being merged. **(F)** Simplified version of $\mathbb{T}$, where I, J, K, and H show four duplication events, K is an ancient duplication happened before G. **(G)** Augmented gene tree T_a^*. **(H)** Augmented gene tree T_b^*.

8: **while** M has more than two tree nodes **do**
9: **if** $u, v \in M$ and u, v and z are connected to an unmapped node x in T_{a_i} **then**
10: MapGeneTreeNode(x)
11: remove u, v and (if necessary) z from M
12: **if** M is empty **then**
13: set M to the set consisting of just the root node r and set $t_r = B$
14: **else**
15: add x to M and set $t_x = t_u \cup t_v \cup \{x\}$
16: **end if**
17: **end if**
18: **end while**
19: **end for**

We now sketch the proof of Theorem 1 as follows. Detailed proof can be found in the Appendix.

Proof. We will show that the algorithm ReconcileTrees produces an isometric reconciliation or detects if no such reconciliation is possible. First, we claim that at every step in the process, each of the maximally internal mapped nodes $m \in M$ in the current gene tree $B = T_{a_i}$ is associated with a subtree t_m of mapped nodes of B that has the following properties

(1) If t_m does not contain the root r then $\Phi(m)$ lies above $\Phi(y)$ for all $y \neq m \in t_m$ and $d(m, y) = D(\Phi(m), \Phi(y))$, else $\Phi(r)$ lies above $\Phi(y)$ for all $y \neq r \in t_m$ and $d(r, y) = D(\Phi(r), \Phi(y))$
(2) All leaves of t_m are mapped to the genome of the species to which they belong

and that Φ is the unique mapping with these properties. This is trivially true after the initial step, in which each of the leaves of B is mapped. We verify that if this is true before a pass through the sequence of operations in the while loop, then it is still true after that sequence of operations, if the input is not rejected. Therefore it is true when the processing of B is complete, if the input is not rejected. However, at this point the set M consists of just the root node r and all nodes of B are included in t_r. Thus, every node x in B will be uniquely mapped to a point $\Phi(x)$ in $\mathbb{T}$ that lies below $\Phi(r)$ such that $d(x, r) = D(\Phi(x), \Phi(r))$. Hence, B will be uniquely isometrically reconciled with $\mathbb{T}$. It follows that if the algorithm ReconcileTrees terminates without rejecting its input, it produces a unique isometric reconciliation of all its input gene trees with its input species tree.

We also show that if the procedure MapGeneTreeNode rejects its input then the tree B is not isometrically reconcilable, it follows that if ReconcileTrees rejects its input then its input is not isometrically reconcilable. Thus, ReconcileTrees is correct. We finally prove that the time complexity of ReconcileTrees can be bounded in $O(nm)$, where n is the number of tree nodes in species tree T and m is the total number of tree nodes in all gene trees. This completes the proof. ■

Once we have the reconciled tree $\mathbb{T}$ and the rooted gene tree T_{a_i}, we could easily embed T_{a_i} into $\mathbb{T}$ according to the mapping Φ for each node. However, for some nodes g in $\mathbb{T}$, we may not have any node x in the original T_{a_i} such that $\Phi(x) = g$. This is because in a single gene family T_{a_i} we usually do not have explicit branching nodes for all the combined speciations and duplications that occur for the gene families in the final $\mathbb{T}$. Therefore, in order to facilitate the gene-order inference in the following sections, we augment T_{a_i} in the following way.

AugmentGeneTree($\mathbb{T}$, T_{a_i}). For each branch (u, v) in T_{a_i}, let $f = \Phi(u)$ and $h = \Phi(v)$ in $\mathbb{T}$. For every node g on the path from f to h in $\mathbb{T}$ excluding f and h, we add a node x on branch (u, v) such that $d(u, x) = D(f, g)$ (and $d(x, v) = D(g, h)$). Set $\Phi(x) = g$. We call the resulting T_{a_i} an **augmented gene tree**, denoted as $\mathsf{T}^*_{a_i}$.

Note that the genes added along the branches of each gene tree represent intermediate forms that are inferred to have existed but do not appear in the original gene tree for that family. Also, the root of an augmented gene tree $\mathsf{T}^*_{a_i}$ does not always have to map to the root of $\mathbb{T}$.

Example. In Figure 2, we originally have a species tree and unrooted gene trees for gene family a and b. We also have the distance information, as shown in Figure 2A–C. We first reconcile T_a and add duplication nodes H and I to $\mathbb{T}$ (Fig. 2D). After T_b is merged, duplication nodes J and K are added (Fig. 2E). Note that based on the distance information, we would know that K corresponds to an ancient duplication before G. Figure 2F shows a simplified version of $\mathbb{T}$ without embedded genes. Figure 2G,H are the rooted augmented gene trees for both gene families.

Direct ancestor and direct descendant. In an augmented gene tree $\mathsf{T}^*_{a_i}$, for every branch (u, v), where u is the parent of v in the tree, let $g = \Phi(v)$ and $h = \Phi(u)$ (based on the result of reconciliation, (g, h) must be an edge in $\mathbb{T}$, where h preceded g):

(1) We call u the **direct ancestor** of v, denoted as $\mathsf{A}_h(g[v]) = u$.
(2) We call v a **direct descendant** of u. The direct descendant may not be unique since u could be duplicated at h, and therefore we denote this as $v \in \mathsf{D}_g(h(u))$.

If a node v in $\mathsf{T}^*_{a_i}$ has no ancestor, then $\mathsf{A}_h(g[v]) = \emptyset$. Conversely, if a node u has no descendant in g, then $\mathsf{D}_g(h[u]) = \emptyset$.

Example. In the augmented atom trees from Figure 2G,H, $\mathsf{A}_E(B[a^1]) = E[a^1]$, $\mathsf{D}_C(J[b]) = \{C[b^1], C[b^2]\}$, $\mathsf{D}_I(G[b^2]) = \emptyset$.

2.4. Reconstructing ancestral adjacencies

After obtaining a reconciled tree $\mathbb{T}$ and augmented gene trees T_{a_i} for all gene families, our goal is to determine lists of gene orders that closely approximate the genome structure of the target ancestral genome, α. We achieve this in two phases:

(1) For each gene x in α, we determine a set of predecessors for x that gives the minimum number of predecessor changes for x based on parsimony. We also determine such a set of successors for x. Then we transform the predecessor and successor relationships in α into potential ancestral adjacencies for each gene in α.
(2) Based on the predicted ancestral adjacencies, we connect the genes into CARs.

We discuss (1) in this section, leaving (2) to the next section. Our approach for (1) is inspired by Fitch's method (Fitch, 1971), which was originally used to infer minimum character changes in a specified tree topology. For that problem, one is given a phylogenetic tree and a letter for every position in each leaf of the tree (corresponding to the contents of orthologous sequence sites). The problem is to infer the ancestral letters (corresponding to internal nodes of the tree), so as to minimize the number of substitutions, i.e., differences between the letters at each end of an edge in the tree.

Here, we deal with sequences of genes having orientations, rather than characters of nucleotides or amino acids, and instead of keeping track of letters at a particular sequence position, we track the genes for both of the immediately adjacent positions. For example, in Figure 3, the leaves indicate that $s(A[a]) = A[b]$, $s(B[a]) = B[c]$, $s(C[a]) = C[b]$, and $s(D[a]) = D[c^1]$. Which gene preceding $G[a]$ would give a minimum number of predecessor changes to explain the data in leaves?

Predecessor set and successor set. For any genome g, we associate with each gene $g[x]$ (including its reverse complement $-g[x]$) two sets of signed genes, denoted $P(g[x])$ and $S(g[x])$ ($P(-g[x])$ and $S(-g[x])$ for reverse complements), giving potential predecessors and successors of $g[x]$. If g is a modern genome, $P(g[x]) = \{p(g[x])\}$ and $S(g[x]) = \{s(g[x])\}$ for each $g[x]$. If x is not in g, then both sets are empty.

We also allow the direct ancestor and direct descendant operations to be applied to a set of genes, i.e., $\mathsf{A}_h(P(g[x])) = \{\mathsf{A}_h(y) \mid y \in P(g[x])\}$, and the result is also a set. $\mathsf{D}_h(P(g[x]))$ is defined analogously.

The inference procedure for the predecessor and successor of each ancestral gene (and its reverse complement) in the augmented gene tree consists of two stages. The first stage works in a bottom-up fashion. The general idea is as follows. For each node x in the augmented gene tree, let g be the genome containing x and let u and v be its two children in the tree. We compute its predecessor set according to

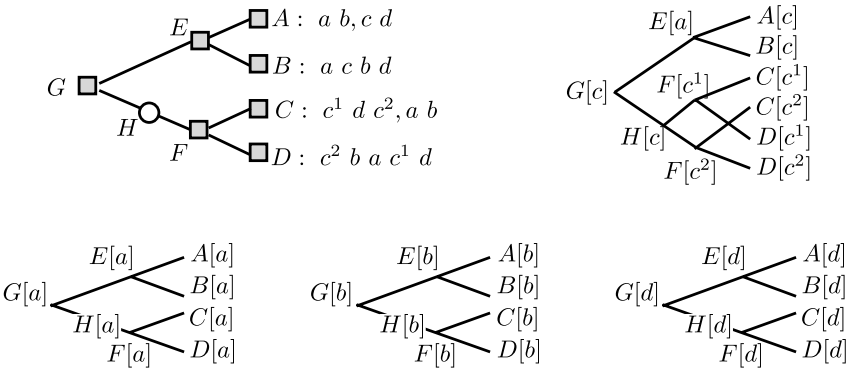

FIG. 3. This figure shows a reconciled tree for four leaf genomes as well as the augmented gene trees for genes a, b, c, and d. Duplication event H only duplicates c. The comma separates two chromosomes in genomes A and C.

the following rule: If x is a leaf, then $P(x)$ consists of the unique predecessor. Otherwise, $P(x)$ is equal to the intersection or union of $\mathsf{A}_g(P(u))$ and $\mathsf{A}_g(P(v))$, depending on whether $\mathsf{A}_g(P(u))$ and $\mathsf{A}_g(P(v))$ are disjoint or not. Here we need to apply the direct ancestor A operation for each gene in a predecessor set, because each gene tree may have a different evolutionary history at the point of g. Similarly, we infer the successor set. The recursive procedure is described in GetPredSuccBottomUp(x). Note that it is possible that x only has one child. We define $P(nil) = \emptyset$, $S(nil) = \emptyset$, and $\mathsf{A}_g(\emptyset) = \emptyset$, then the calculations from line 5 to 14 still hold when x only has one child in the augmented gene tree.

In the second phase, we propagate the information down the tree so that $P(x)$ and $S(x)$ for every gene x in the target ancestor α can incorporate outgroup information. It works in a top-down fashion. For each node x in the augmented gene tree, let w be x's parent in the tree. We propagate $P(w)$ down the tree to adjust the original $P(x)$. We also need the direct descendant operation D here. $S(x)$ is adjusted analogously. The whole procedure is summarized as AdjustPredSuccTopDown(w, x) in recursive form.

GetPredSuccBottomUp(x).

1: **if** x is non-leaf node **then**
2: GetPredSuccBottomUp(u)
3: GetPredSuccBottomUp(v)
4: $g \Leftarrow \Phi(x)$
5: **if** $\| \mathsf{A}_g(P(u)) \cap \mathsf{A}_g(P(v)) \| \neq 0$ **then**
6: $P(x) \Leftarrow \mathsf{A}_g(P(u)) \cap \mathsf{A}_g(P(v))$
7: **else**
8: $P(x) \Leftarrow \mathsf{A}_g(P(u)) \cup \mathsf{A}_g(P(v))$
9: **end if**
10: **if** $\| \mathsf{A}_g(S(u)) \cap \mathsf{A}_g(S(v)) \| \neq 0$ **then**
11: $S(x) \Leftarrow \mathsf{A}_g(S(u)) \cap \mathsf{A}_g(S(v))$
12: **else**
13: $S(x) \Leftarrow \mathsf{A}_g(S(u)) \cup \mathsf{A}_g(S(v))$
14: **end if**
15: **end if**

AdjustPredSuccTopDown(w, x).

1: **if** w is non-leaf node **then**
2: **if** w is not the root **then**
3: $h \Leftarrow \Phi(w)$, $g \Leftarrow \Phi(x)$
4: **if** $\| \mathsf{A}_h(P(x)) \cap P(\mathsf{A}_h(x)) \| \neq 0$ **then**
5: $P(x) \Leftarrow P(x) \cap \mathsf{D}_g(\mathsf{A}_h(P(x)) \cap P(\mathsf{A}_h(x)))$
6: **end if**
7: **if** $\| \mathsf{A}_h(S(x)) \cap S(\mathsf{A}_h(x)) \| \neq 0$ **then**
8: $S(x) \Leftarrow S(x) \cap \mathsf{D}_g(\mathsf{A}_h(S(x)) \cap S(\mathsf{A}_h(x)))$
9: **end if**
10: AdjustPredSuccTopDown(x, u)
11: AdjustPredSuccTopDown(x, v)
12: **end if**
13: **end if**

The time complexity for both procedures is $O(mn)$, where m is the number of nodes in the augmented gene tree and n is the total number of leaves in the augmented gene tree, because the tree traversal takes $O(m)$ and there are at most n members in each set, either P or S.

Theorem 2. *For any augmented gene tree,* GetPredSuccBottomUp *and* AdjustPredSuccTopDown *will give a set of genes for each node in the tree that are most parsimonious in terms of successor changes in the entire augmented gene tree. Similarly, it will also give a set of genes for each node that are most parsimonious in terms of predecessor changes.*

Proof. We first prove that GetPredSuccBottomUp gives a set of genes to node x that are most parsimonious in terms of successor changes to the descendants of x. Let $k(x)$ denote the minimum number of successor changes in the subtree rooted at x. We prove the above claim by induction. Let u and v be the two children of x. *Basis:* if tree height $h = 1$, then u and v are leaves in the gene tree. If u and v are the same, then no successor change is needed; $k(x) = 0$. Otherwise, only 1 change is needed based on the bottom-up inference; $k(x) = 1$. *Induction:* we assume the above claim is correct when the subtree height is h, then prove it for height $h + 1$. If the intersection of successor sets for u and v is not empty, then we can have $k(x) = k(u) + k(v)$ by assigning any gene in the intersection to x. Otherwise, the optimal $k(x)$ is $k(u) + k(v) + 1$, by assigning any gene in the union of successor set from u and v.

We then prove that AdjustPredSuccTopDown gives a set of genes that will have the most parsimonious successor changes from x's ancestors to x without changing the parsimony scores k for each node. Let $j(x)$ denote the minimum number of successor changes from the root of the augmented gene tree to x. Let w be the parent node of x. *Basis:* when path length $h = 1$ from the root to x, i.e. w is the root, if w and x share the same successors, then no successor change is needed if we assign both w and x a successor they share; $j(x) = 0$. Otherwise, only 1 change is needed from any gene of the successor set of w to any gene of the successor set of x; $j(x) = 1$. *Induction:* we assume the above claim is correct when the path length from root is h, then prove it for length $h + 1$. If the intersection of successor sets for w and x is not empty, then we can have $j(x) = j(w)$ by assigning any gene in the intersection to x. Otherwise, the optimal $j(x)$ is $j(w) + 1$. Note that the adjustment performed by the AdjustPredSuccTopDown does not change the parsimony score $k(x)$, i.e. the final genes in the successor set for x still gives the most parsimonious successor changes in its subtree. This completes the proof. We can prove in the same way that the claim is also true for predecessor changes. ■

Example. In Figure 3, after running the above algorithm, we will have $S(G[a]) = \{G[b], G[c]\}$. We also will have $P(G[c]) = \$$, etc.

In the above example, it is interesting to note that even though $G[c] \in S(G[a])$, $G[a] \notin P(G[c])$.

Ancestral adjacency graph. As shown in the previous example, we know that $x \in S(y)$ does not guarantee that $x \in P(y)$, and vice versa. However, in order to get the ancestral adjacency, we need to retain consistent predecessor and successor relationships in the target ancestor. We achieve this by constructing an ancestral adjacency graph.

We first construct a directed edge set E_1 such that: $E_1 = \{(y, x) \mid y \in P(x)\}$, for x and its reverse complement $-x$ in the target ancestor. Here (y, x) denotes an arc directed from y, a predecessor of x, to x. We also construct directed edge set E_2 such that: $E_2 = \{(x, y) \mid y \in S(x)\}$. Here (x, y) denotes an arc directed from x to y, a successor of x.

We then construct a digraph $G = (V, E)$, where:

$$V(G) = \{x, -x \mid x \in \alpha, x \neq \$\}$$

$$E(G) = \{E_1 \cap E_2\}$$

Node set $V(G)$ includes all the genes and their reverse complements, i.e., nodes exist in gene pairs as x and $-x$ in the target ancestor α. Therefore, $|V| = 2N$, where N indicates the total number of genes in the target ancestor. The intersection of E_1 and E_2 will eliminate all the edges where one of the endpoints is \$ because there are no predecessor and successor sets for \$. Since there is no \$ in G, the implication for the following process of finding CARs is that when a gene can be followed most parsimoniously by either certain other genes or \$, we will favor the non-\$ to reduce the number of CARs. (Similarly for predecessors.)

Now, we have the ancestral adjacency graph G indicating consistent predecessor and successor relationships that are supported by $\mathbb{T}$, all $\mathsf{T}^*_{a_i}$, and the leaf genomes.

2.5. From ancestral adjacency to ancestral gene order

The edges of the ancestral adjacency graph G indicate consistent predecessor and successor relationships. However, they do not necessarily indicate a unique adjacency relationship for a particular gene. Three

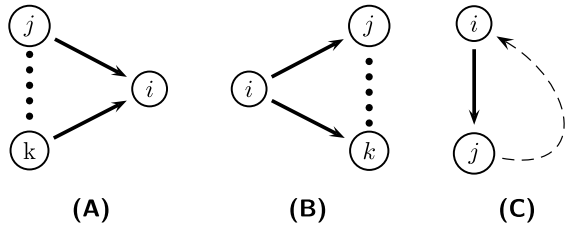

FIG. 4. Three potential ambiguous cases in the ancestral adjacency graph G.

potential ambiguous cases can occur in G, as depicted for node i in Figure 4. In (A), i has several incoming edges; in (B), i has several outgoing edges; in (C), i forms a cycle with j, where each node j satisfies indegree(j) = outdegree(j) = 1. (If a more complex cycle exists, then some node falls in either case (A) or case (B)).

Problem. In our directed cyclic graph $G(V, E)$, each edge (i, j) has a weight w_{ij}. If no weights are given, we set every w_{ij} to 1. Our objective is to find a set of vertex-disjoint directed paths that covers all the nodes in G and at the same time maximizes the sum of weights in the resulting paths. We call this the maximum weighted path cover problem (MWPCP). We also allow degenerate paths consisting of a node without edges. We call each resulting path a CAR.

If none of the ambiguous cases (Fig. 4) are present, then G itself forms the set of paths that covers all the nodes. In this case, the CARs can be directly defined from this graph as discussed below. When ambiguity exists, we need to resolve it and choose appropriate directed edges to form CARs.

The case where all w_{ij} are 1 is equivalent to the minimum path cover problem (MPCP), i.e., finding the minimum number of vertex-disjoint paths covering all the nodes in the digraph. In that case, the result from MPCP is also the result for MWPCP in a unit-edge-weight graph, because the minimum number of paths implies the maximum number of edges. The MPCP is **NP**-hard (Boesch and Gimpel, 1977). When antiparallel edges are not allowed, i.e., directed edges (i, j) and (j, i) do not co-exist, there are approximation algorithms for MWPCP that have been studied (Moran et al., 1990).

We use a greedy approach to achieve an approximate solution when antiparallel edges are allowed, given in the algorithm of FindCARs below. We first sort the edges by weight. Then the greedy approach always tries to add the heaviest edge to the resulting path set.

FindCARs(G).

Input: ancestral adjacency graph G
Output: contiguous ancestral regions in G

sort edges by weight in descending order.
create a new graph C, $V(C) \Leftarrow V(G)$ and $E(C) \Leftarrow \emptyset$
for all edge $(i, j) \in E(G)$ **do**
  $h(i) \Leftarrow i$ and $q(i) \Leftarrow i$
  $h(j) \Leftarrow j$ and $q(j) \Leftarrow j$
end for
for all available $(i, j) \in E(G)$, in order of edge weight **do**
  if outdegree(i) = 0 and indegree(j) = 0 and $h(i) \neq h(j)$ **then**
    add edge (i, j) and $(-j, -i)$ to $E(C)$
    update outdegree(i) and indegree(j) in C
    update outdegree($-j$) and indegree($-i$) in C
    $h(q(j)) \Leftarrow h(i)$, $q(q(i)) \Leftarrow q(j)$, $q(q(j)) \Leftarrow q(i)$
    $h(q(-i)) \Leftarrow h(-j)$, $q(q(-i)) \Leftarrow q(-j)$, $q(q(-j)) \Leftarrow q(-i)$
  end if
end for

Note that h and q in the above algorithm are two auxiliary variables that are used to avoid creating cycles. $h(i)$ indicates the path that node i is in and $q(i)$ is the other endpoint of the path. The requirement on line 8 ($h(i) \neq h(j)$) guarantees that there will be no cycle in the result path cover. The remaining paths in C correspond to the desired CARs.

When adding edges into an existing path, particular care is needed to avoid putting j and $-j$ in the same CAR. We also add both (i, j) and its symmetric version, $(-j, -i)$. For each path found by this approach, a symmetric path in the opposite orientation is also found, since we have nodes for both i and $-i$. The two paths correspond to the same CAR, and eventually we choose one of them.

Letting G have n edges, the time complexity of FindCARs is $O(n \log n)$ because the sorting takes $O(n \log n)$, the initialization from lines 3–6 takes $O(n)$, and lines 7–15 take $O(n)$.

Theorem 3. FindCARs *is a factor-3 approximation algorithm for* MWPCP.

Proof. Line 8 in FindCARs guarantees that the result cover is a path cover of G. Let M^* be the optimal path cover for MWPCP and let FindCARs generate M. We consider each arc (b, c) in M^*. If (b, c) is not in M, there are three possible cases:

(1) Based on the greedy approach, there is at least one of (b, e) or (f, c) that is in M (Fig. 5A) such that $w_{bc} \leq \max(w_{be}, w_{fc})$. But since M^* is optimal, we have $w_{bc} \geq w_{be} + w_{fc} \geq \max(w_{be}, w_{fc})$. Therefore, this case happens only when $w_{bc} = \max(w_{be}, w_{fc})$ and $\min(w_{be}, w_{fc}) = 0$. In other words, this does not change the total weight between M^* and M.

(2) There are (e, b) and (c, f) in M and there is a path from c to b going through e and f in M with additional two possible edges (a, b) and (c, d) in M^* but not in M (Fig. 5B). Note that e and f could be the same node. The worst case for this situation is that edges (e, b) and (c, f) can be replaced by edges $(a, b), (b, c), (c, d)$ to have a better cover. In this case, based on the greedy approach, we have $w_{ab} + w_{bc} + w_{cd} \leq w_{eb} + w_{cf} + \min(w_{eb}, w_{cf}) = 2\min(w_{eb}, w_{cf}) + \max(w_{eb}, w_{cf})$. Since M^* is optimal, we also have $w_{ab} + w_{bc} + w_{cd} \geq w_{eb} + w_{cf} = \min(w_{eb}, w_{cf}) + \max(w_{eb}, w_{cf})$, otherwise we would get the contradictory result such that we could have a cover better than M^* by removing (a, b), (b, c), and (c, d) in M^* and adding the path from c to b going through e and f. Therefore, $w_{eb} + w_{cf} \geq \frac{2}{3}(w_{ab} + w_{bc} + w_{cd})$.

(3) As shown in Figure 5C, this case is similar to (2), but there is only (c, b) in M, i.e., there is a pair of antiparallel edges. For the worst case, similar to the deduction in (2), we can bound the difference such that $w_{cb} \geq \frac{1}{3}(w_{ab} + w_{bc} + w_{cd})$.

Therefore, in the worse case, every edge (c, b) in M produced by FindCARs can be replaced by three edges (a, b), (b, c), and (c, d) in M^* to get a better cover. Hence, $\sum_{(i,j)\in M} w_{ij} \geq \frac{1}{3}\sum_{(i,j)\in M^*} w_{ij}$. ■

Finally, we describe a simple approach to determine the edge weights w_{ij}. For a directed edge (i, j), if outdegree$(i) = 1$ and indegree$(j) = 1$, we set $w_{ij} = 1$. Otherwise, the corresponding weight $w_{ij} = w_\alpha(i, j)$ (α is target ancestor) is determined recursively.

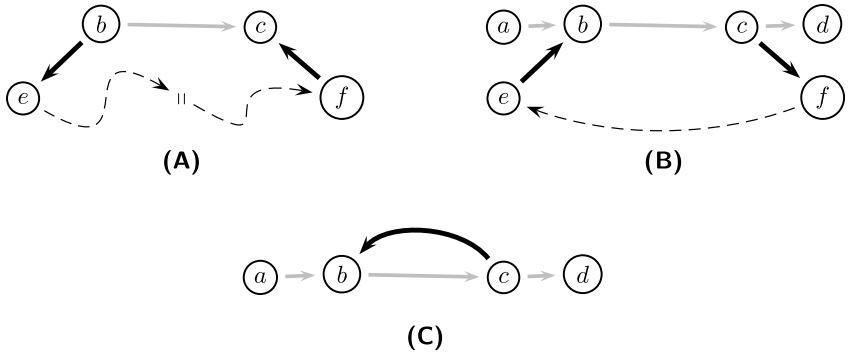

FIG. 5. Three possible cases when the arc (b, c) is in the optimal solution but not in the CARs produced by FindCARs.

Before computing $w_\alpha(i, j)$, we modify the reconciled tree $\mathbb{T}$ by rerooting it if α is not the root of $\mathbb{T}$. We denote the rerooted binary tree as $\mathbb{T}'$. Then we apply the following rule:

$$w_\alpha(i, j) = \frac{D(\alpha, \tau) \cdot w_\varphi(i, j) + D(\alpha, \varphi) \cdot w_\tau(i, j)}{D(\alpha, \tau) + D(\alpha, \varphi)}$$

where $i = a_x^m$ and $j = a_y^n$, $D(\alpha, \tau)$ and $D(\alpha, \varphi)$ are the branch lengths from α to the first left and first right non-duplication node in $\mathbb{T}'$; $w_\tau(i, j)$ and $w_\varphi(i, j)$ are the edge weights on τ and φ, respectively. On a leaf genome f of $\mathbb{T}'$, if there is an adjacency (s, t) present in f such that s belongs to gene family a_x and t belongs to gene family a_y, we set $w_f(i, j) = 1$, otherwise $w_f(i, j) = 0$. Note that if an edge (i, j) is involved in ambiguous case (a) or (b), then $w_{ij} < 1$. The underlying assumption of the above equation is that rearrangement is more likely to happen on longer branches.

2.6. The summary of the DUPCAR algorithm

In outline, the whole DUPCAR algorithm can be described as follows, where α is the target ancestor.

DUPCAR($\mathbb{G}, T, \alpha, \mathsf{T}_{a_i, 1 \le i \le N}$).

Input: modern genomes $\mathbb{G}$, species tree T, target ancestor α, gene trees T_{a_i}
Output: contiguous ancestral regions in α

1: ReconcileTrees(T, T_{a_i})
2: initialize $P(x)$ and $S(x)$ for each gene x in $\mathbb{G}$.
3: **for all** gene families a_i **do**
4: GetPredSuccBottomUp($\mathsf{T}^*_{a_i}$)
5: AdjustPredSuccTopDown($\mathsf{T}^*_{a_i}$)
6: **end for**
7: build the ancestral adjacency graph G for α.
8: FindCARs(G)

3. RESULTS

3.1. Simulations

We used extensive simulations to validate our analysis. The simulator starts with a hypothetical "ancestor" genome, which evolves into the extant species through speciation and operations including inversion, translocation, fusion, fission, insertion, deletion, and duplication. When an operation is applied, breakpoints are chosen uniformly at random from the set of used or unused breakpoints on that chromosome, according to the breakpoint reuse ratio.

We tuned the weights of these operations in order to generate simulated data that closely resemble placental mammalian genomes. We simulated datasets using the phylogenetic tree: `((((human,chimp),rhesus),(mouse,rat)),dog)`. The ancestor genome was assigned 2,000 genes. After evolution is performed, we collapse genes where order and orientation are conserved across all species. For instance, if genes a, b, and c appear in every genome with order $(a\ b\ c)$, then we treat these three genes as one. This strategy allows us to evaluate only the varied adjacencies, since the unbroken adjacencies will be found by essentially any procedure.

The simulator's parameters or weights of the large-scale operations were adjusted to guarantee that the extant species had approximately 1000 genes after collapsing. Moreover, we arranged that approximately 10% of the genes are in multi-gene families.

We generated synthetic data in two categories: (1) data without insertion and deletion of genes and (2) data with insertion and deletion of genes, where the insertion and deletion ratio is 1:2. For each category, we simulated data by varying the breakpoint reuse ratio (0%, 10%, 20%, 30%, 40%, and 50%), with 100 datasets per parameter set. The breakpoint reuse ratio is defined as the percentage of all ancestral adjacencies that were changed more than once during the simulated evolution.

We ran our DUPCAR reconstruction program for inferring CARs on each dataset (average running time of 1.5 min) and compared the predicted adjacencies with the real (simulated) ones. We calculate the error rate, S, of adjacencies:

$$S = \frac{|R \cup P| - |R \cap P|}{|R \cup P|} \times 100\%$$

where R is the set of adjacencies in the real genome, P is the adjacencies in the predicted genome, and $|X|$ denotes the size of the set X. S represents the percentage of adjacencies where the real genome and the predicted genome disagree.

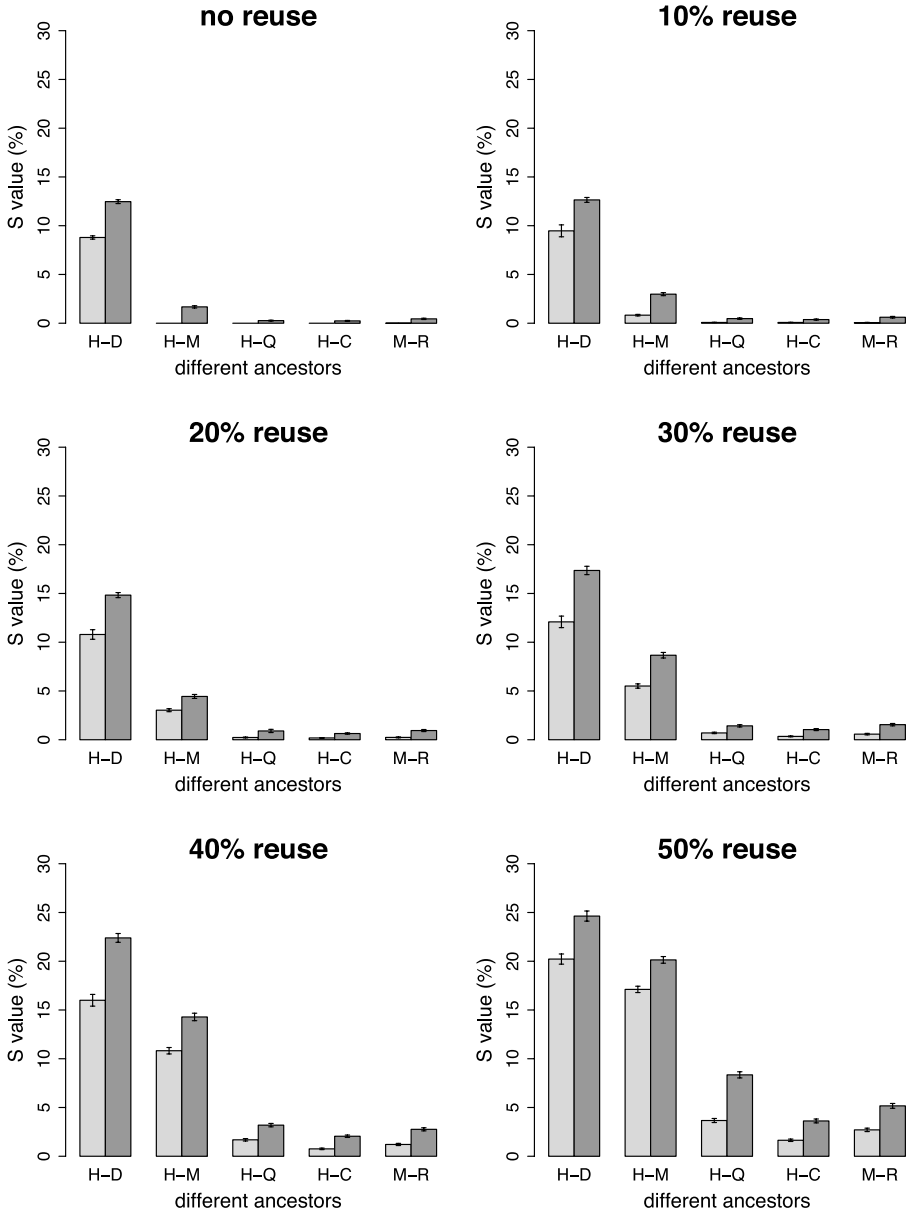

FIG. 6. The error rate, S (defined in the main text), for the reconstruction of each ancestral genome when we vary the breakpoint reuse ratio. The light gray bars represent data without insertion and deletion of genes, while darker bars represent data with indels. 95% confident intervals for the error rate of every parameter set are also shown. (H-D, human-dog ancestor; H-M, human-mouse ancestor; H-Q, human-rhesus ancestor; H-C, human-chimp ancestor; M-R, mouse-rat ancestor.)

We compared every ancestral genome, including the human-dog ancestor, which has no outgroup information. The results are summarized in Figure 6. For every ancestor, the error rate increases when the breakpoint reuse ratio increases. Also, the error rate for category (1) (without indels) is always less than for category (2) (with indels). In the first picture in Figure 6, when we do not allow breakpoint reuse (i.e. each ancestral adjacency has been broken at most once), S is 0 for category (1) for each ancestor except the human-dog ancestor, indicating 100% accuracy. We could not attain 100% accuracy for the human-dog ancestor without an outgroup to resolve ambiguities.

The insertions and deletions hurt the accuracy, mostly because we only infer the presence or absence of a gene in an ancestral genome directly from the gene trees that are inferred based on what we observe from the leaves. For example, if gene family x is present in both primates and rodents but absent in dog, we will simply root the gene tree under the human-dog ancestor, and will never reconstruct x in the human-dog ancestor. However, it is possible that gene x was deleted on the dog lineage.

3.2. Application to mammalian chromosome X reconstruction

We compared genomic sequence of the X chromosomes of four species (human, mouse, rat, and dog), partitioning the chromosome into 201 conserved segments that haven't been disrupted by rearrangements or duplications using blastz pairwise alignments and self-alignments (Schwartz et al., 2003), covering over 85% of the human genome. The species tree is `(((hg18,(mm8,rn4)),canFam4)`, where leaf names indicate the assemblies we used. For each family, multiple alignments are created and a distance matrix is calculated using the Jukes-Cantor model. A gene tree, or more precisely a conserved segment tree, is inferred using a modified version of the Neighbor-Joining algorithm, which utilizes the genomic context of the genes in addition to evolutionary distances in the tree-building process. Duplications and losses are identified in the gene trees via reconciliation with the species tree. All of the duplication events are combined into a single reconciled tree, and augmented gene trees are generated accordingly.

We applied our new algorithm to the chromosome X data and successfully reconstructed the boreoeutherian chromosome X into two CARs. The two CARs are separated because of two duplications that happened in the same region, confusing the adjacency-inference procedure. Other than that, the result is quite consistent with previous reconstructions, in the sense that human chromosome X has experienced no large-scale rearrangements since the *boreoeutherian* ancestor, except a few in-place inversions. In the results of Murphy et al. (2005), which were computed with MGR, there was no rearrangement between human chromosome X and the *boreoeutherian* chromosome X. In Ma et al. (2006), there were four predicted inversions in human. With our current data and the new algorithm, we found five human inversions. Our two reconstructions agree on the three inversions shown in Table 1.

3.3. Application to a ciliate genome

It has been shown that the genome of the unicellular eukaryote *Paramecium tetraurelia*, a ciliate with roughly 40,000 genes, resulted from at least three whole-genome duplication (WGD) events, with additional

TABLE 1. COMPARISON BETWEEN THE RECONSTRUCTED *boreoeutherian* CHRX USING DUPCAR AND DATA FROM MA ET AL. (2006)

	No. of segs	*Coverage of hg18*	*No. of duplication events*	*Human rearrangements*
Ma et al. (2006)	90	91%	0	chrX:16164900-16249687 chrX:73544925-73628542 chrX:126526019-126592597 chrX:139946312-140388068
DUPCAR	201	87%	26	chrX:16166900-16235111 chrX:73547455-73672237 chrX:108284817-108330640 chrX:109709282-109765848 chrX:139961357-140378962

TABLE 2. THE NUMBER OF CARS RECONSTRUCTED IN THREE TARGET ANCESTRAL GENOMES

Target ancestor	*Genes we included*	*anc genes with paralog from Aury et al. (2006)*	*Gene families we used*	*Predicted CARs*	*anc blocks in Aury et al. (2006)*
Old WGD	2,981	1,530	559	125 (75)	43
Intermediary WGD	11,620	7,996	3,118	266 (126)	81
Recent WGD	25,708	24,052	9,951	442 (180)	131

In the column of predicted CARs, the number of CARs with more than two genes is given in parentheses.

rearrangement operations (Aury et al., 2006). Those authors reconstructed the genome architectures of four ancestral genomes, corresponding to the most recent WGD, the intermediary WGD, the old WGD, and the ancient WGD. They used Best Reciprocal Hits to construct a paralogon, which is a pair of paralogous blocks that could be recognized as derived from a common ancestral region. Then paralogons were merged into single ancestral blocks, and the process iterated until reaching the ancient WGD. However, they did not intend to determine the gene orders in each ancestral block; when a paralogon was constructed, the detailed order and orientation of genes inside the block were ignored.

The 39,642 genes form 22,635 gene families (including 11,740 single-gene families), spread over 676 scaffolds in the current genome assembly. We tested our algorithm by reconstructing all WGDs except the ancient WGD. We used the gene order in modern *Paramecium tetraurelia* and the gene trees from Aury et al. (2006). The reconciled tree is special in this case, which contains one leaf genome (the modern one), as well as ancestral nodes representing duplication events. We built the augmented gene trees accordingly.

Many genes do not have a paralog in the paralogons for a particular ancestral genome. If we were to include all the gene families in the reconstruction, the input data would be very noisy and the resulting CARs would be too fragmented, due to the fact that we only have one leaf genome. Therefore, when reconstructing CARs in a certain target genome, we did some preprocessing to retain only genes that have paralogs derived from ancient duplications. Additional genes were added if their paralogs (from this duplication) were retained in the leaf genome.

For all three genomes, the number of CARs reconstructed by us is greater than the number of ancestral blocks reported in Aury et al. (2006) using the paralogon method (Table 2). There are two reasons for this: (1) Aury et al. (2006) ignored gene orders, whereas we take order and orientation into account when inferring CARs. (2) We used more genes in the reconstruction than just the ancestral genes with paralogs, which were essentially used as anchors when building paralogons.

In general, our prediction is a refinement of the result of Aury et al. (2006). In Figure 7, we show the mapping from our CARs (126 CARs containing more than two genes) for the intermediary WGD to the ancestral blocks constructed using the paralogon method. Most of them follow the pattern that several CARs correspond to one ancestral block of Aury et al. (2006), indicating an agreement in regional mapping if we ignore the gene orders. The mappings of reconstructions for the old WGD and the recent WGD have similar results (data not shown). Since Aury et al. did not reconstruct the ancestral gene adjacencies, we could not compare our prediction with theirs in detail.

Recent approaches to the genome halving problem (Seoighe and Wolfe, 1998; El-Mabrouk and Sankoff, 2003; Alekseyev and Pevzner, 2007; Zheng et al., 2006) might be particularly useful and interesting if applied to this ciliate genome. As more ciliate genomes become available, we plan to further investigate the changes of gene orders between different WGDs, using outgroup information from the new genomes to identify additional adjacencies, which will help determine which methods of reconstructing ancestral architecture are best, and may shed more light on *Paramecium* evolution.

4. DISCUSSION

In this paper, we extend the method in Ma et al. (2006) and propose an algorithm, called DUPCAR, to reconstruct ancestral gene orders with duplications. Most previous gene-order reconstruction methods have relied on a particular distance metric with restricted set of operations, for example, reversal distance without

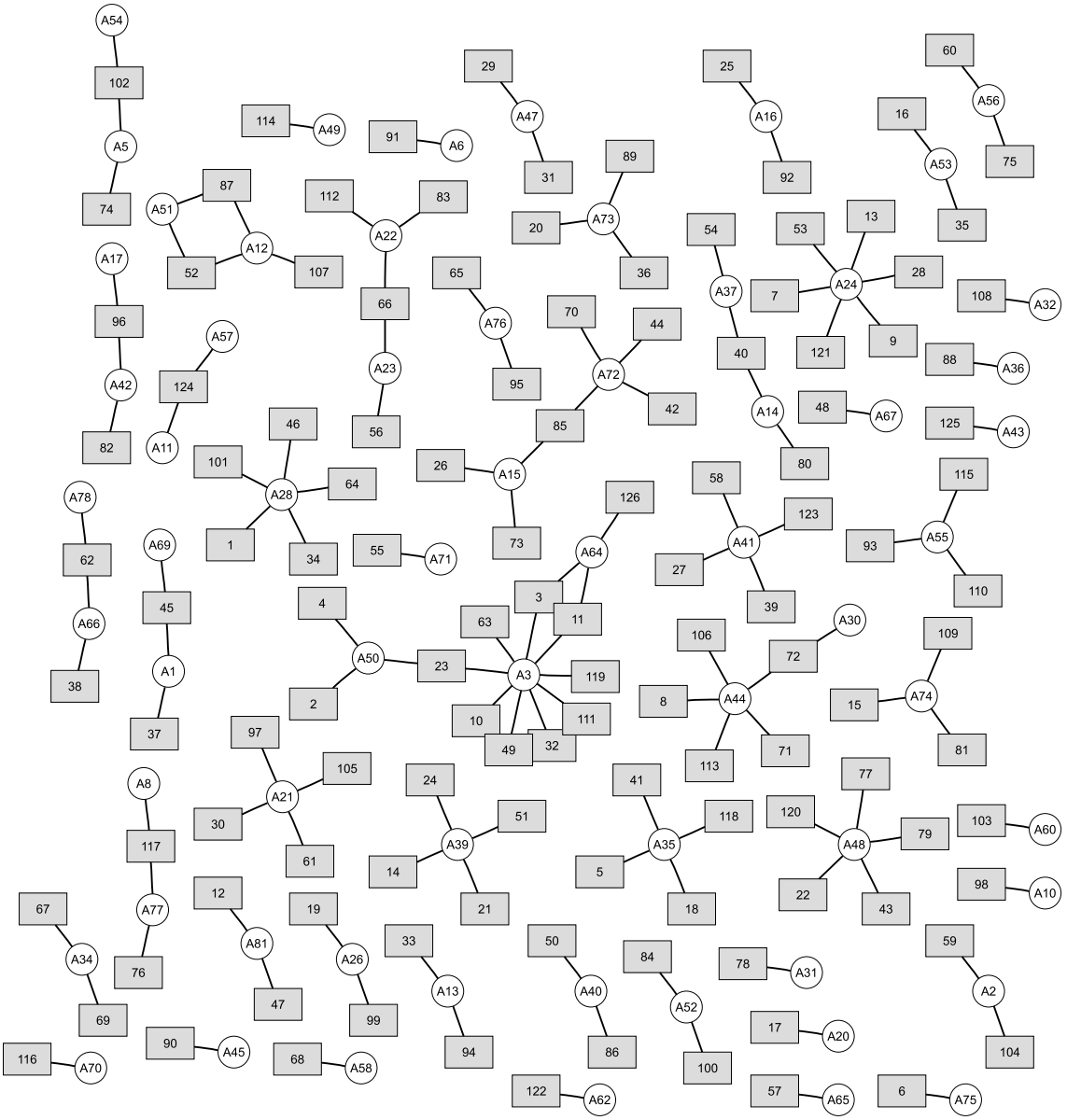

FIG. 7. This figure shows the mapping the CARs of the intermediary WGD predicted using DUPCAR to the ancestral blocks from Aury et al. (2006). Numbers in circles correspond to the ancestral block ID predicted by Aury et al.. Numbers in gray rectangles are the CAR ID.

indels and duplications (Bourque and Pevzner, 2002), breakpoint distance without indels and duplications (Moret et al., 2001), reversal distance with duplications but without indels and translocations (El-Mabrouk, 2002). Our approach avoids to use reversal distance or breakpoint distance and relies on simple parsimony, making it easier to extend to handle different operations.

Because the critical procedure in DUPCAR is based on parsimony, the method retains a relatively small number of possibilities that are equally parsimonious in the ancestor. For the purpose of obtaining fewer CARs, a more reliable probabilistic rearrangement and duplication model might be more appropriate to reconstruct ancestral adjacencies.

We have a simplifying assumption in this paper that all the distances in the gene trees are perfect, which makes it easy to reconcile gene trees to the species tree. In reality, we usually have approximate distances. In our chromosome X experiment, since our resolution was relatively low, the gene trees were built with

strong evidence of genomic distance and context. However, it remains a challenge for high-resolution reconstructions. Therefore, a more robust gene tree reconstruction and reconciliation method is needed (Chen et al., 2000; Bansal et al., 2007; Rasmussen and Kellis, 2007). This is a key area for further work.

Our future work will also focus on incorporating the ability to reconstruct evolutionary history with large-scale operations, instead of just figuring out the gene orders. Although solving the median problem is algorithmically challenging, it is completely feasible to provide a plausible history of rearrangements and duplications on each branch in the phylogeny when both the descendant genome and the ancestor genome have been predicted.

Our simulation of large-scale mammalian genome evolution looks promising. However, our reconstruction of the mammalian chromosome X is in relatively low resolution. In highly rearranged and duplicated regions, especially tandem-duplication-rich regions, special algorithms to figure out recent duplications first would be useful (Zhang et al., 2008).

A number of challenges remain before the genome structure of mammalian ancestors can be accurately predicted in terms of rearrangements and duplications, among which the most difficult would be partitioning the genomes in finer resolution and accurately dating the duplication events with noisy data.

5. APPENDIX

5.1. The proof of Theorem 1

Given a set $T_{a_1}, \ldots, T_{a_N}$ of unrooted gene trees with designated leaf species and a species tree T, the isometric reconciliation of $T_{a_1}, \ldots, T_{a_N}$ with T is unique and there is an efficient procedure to either construct it or detect that no such isometric reconciliation exists.

Proof. We will show that the algorithm ReconcileTrees produces an isometric reconciliation or detects if no such reconciliation is possible.

(A) We prove that if the algorithm accepts the tree then the tree is reconcilable and the reconciliation is unique.

First, we prove that at every step in the process, each of the maximally internal mapped nodes $m \in M$ in the current gene tree $B = T_{a_i}$ is associated with a subtree t_m (constructed in ReconcileTrees) of mapped nodes of B that has the following properties:

(1) If t_m does not contain the root r then $\Phi(m)$ lies above $\Phi(y)$ for all $y \neq m \in t_m$ and $d(m, y) = D(\Phi(m), \Phi(y))$. Else, if t_m contains the root r then $\Phi(r)$ lies above $\Phi(y)$ for all $y \neq r \in t_m$ and $d(r, y) = D(\Phi(r), \Phi(y))$.
(2) All leaves of t_m are mapped to the genome of the species to which they belong

and that Φ is the unique mapping with these properties.

We prove this by structural induction. It is clear that the above claim is true after the initial step, in which each of the leaves of B is mapped. Let m be an unmapped internal node of the gene tree, with branches to nodes u, v, and z. Suppose the mappings of u and v to the species tree have already been determined and we have subtrees t_u and t_v that satisfy the above claim.

We first consider the case that both t_u and t_v do not contain root r. If the tree has not been rooted after mapping m, then $\Phi(m)$ will be mapped to a point in T that is above $\Phi(u)$ and $\Phi(v)$. Based on the induction, $\Phi(m)$ lies above $\Phi(y)$ for all $y \in t_u$ and $y \in t_v$. Hence, $\Phi(m)$ lies above $\Phi(y)$ for all $y \in t_m$. Since we have

$$D(\Phi(m), \Phi(u)) = d(m, u) \quad \text{and} \quad D(\Phi(m), \Phi(v)) = d(m, v),$$

therefore, for all $y \in t_u$, we have

$$D(\Phi(m), \Phi(y)) = D(\Phi(m), \Phi(u)) + D(\Phi(u), \Phi(y)) = d(m, u) + d(u, y) = d(m, y)$$

Similarly, for all $y \in t_v$, we have

$$D(\Phi(m), \Phi(y)) = D(\Phi(m), \Phi(v)) + D(\Phi(v), \Phi(y)) = d(m, v) + d(v, y) = d(m, y)$$

When the gene tree can be rooted after mapping m using u and v, without loss of generality, suppose r is on the path from m to v. Then $\Phi(r)$ will be mapped to a point in T that is above $\Phi(m)$ and $\Phi(v)$. Based on the induction, $\Phi(r)$ lies above $\Phi(y)$ for all $y \in t_m$. For all $y \in t_u$,

$$D(\Phi(r), \Phi(y)) = D(\Phi(r), \Phi(u)) + D(\Phi(u), \Phi(y)) = d(r, m) + d(m, u) + d(u, y) = d(r, y)$$

For for all $y \in t_v$,

$$D(\Phi(r), \Phi(y)) = D(\Phi(r), \Phi(v)) + D(\Phi(v), \Phi(y)) = d(m, v) - d(m, r) + d(v, y) = d(r, y)$$

We then consider the case that one of t_u and t_v contains root r. Without loss of generality, suppose t_v contains the root r. After the mapping of m, $\Phi(m)$ will be mapped to a point on the path from $\Phi(v)$ to $\Phi(u)$ such that $\Phi(m)$ lies under $\Phi(v)$ and $\Phi(u)$ lies under $\Phi(m)$. Based on the induction, $\Phi(r)$ lies above $\Phi(m)$ and $\Phi(y)$ for all $y \in t_v$ and $y \in t_u$. Hence, $\Phi(r)$ lies above $\Phi(y)$ for all $y \in t_m$. Again, based on MapGeneTreeNode, we have

$$D(\Phi(m), \Phi(u)) = d(m, u) \quad \text{and} \quad D(\Phi(m), \Phi(v)) = d(m, v)$$

Therefore,

$$D(\Phi(r), \Phi(m)) = D(\Phi(r), \Phi(v)) + D(\Phi(v), \Phi(m)) = d(r, v) + d(v, m) = d(r, m)$$

Also, for all y in the original t_u,

$$\begin{aligned} D(\Phi(r), \Phi(y)) &= D(\Phi(r), \Phi(v)) + D(\Phi(v), \Phi(m)) + D(\Phi(m), \Phi(u)) + D(\Phi(u), \Phi(y)) \\ &= d(r, v) + d(v, m) + d(m, u) + d(u, y) = d(r, y) \end{aligned}$$

Finally we consider the case where $\Phi(z)$ has also been determined before we map m. There are two possible situations where (i) either t_m or t_z contain the root and (ii) both t_m and t_z do not contain the root at this point. We can use the similar logic above to show that $\Phi(r)$ lies above $\Phi(y)$ for all $y \in t_m$ and $y \in t_z$, and that for all y in t_m and all y in t_z

$$D(\Phi(r), \Phi(y)) = d(r, y).$$

Since the above properties (1) and (2) hold at every step in the process, they also hold when the processing of B is complete, if the input is not rejected. However, at this point the set M consists of just the root node r and all nodes of B are included in t_r. Thus, every node x in B will be uniquely mapped to a point $\Phi(x)$ in T that lies below $\Phi(r)$ such that $d(x, r) = D(\Phi(x), \Phi(r))$. Hence, B will be uniquely isometrically reconciled with T. It follows that if the algorithm ReconcileTrees terminates without rejecting its input, it produces an isometric reconciliation of all its input gene trees with its input species tree.

(B) We prove that if the algorithm rejects the gene tree then the gene tree is not reconcilable.

We now verify that if the above procedure rejects B, then B is not reconcilable. In the algorithm MapGeneTreeNode, we use node u and v in the gene tree B to map the unmapped node m. From the above proof, we know that subtrees t_u and t_v have been uniquely mapped to the T.

In step (5) and step (7)(a) in MapGeneTreeNode, if one tries to root the gene tree B again when B is already rooted when mapping m, then B is certainly not reconcilable because all the nodes that have already been mapped to T are uniquely mapped based on the proof in (A).

We then consider the case in step (1) in MapGeneTreeNode. Because a successful reconciliation maps the nodes of the gene tree to the nodes and edges of the species tree, there will be no shorter path between two mapped nodes in the gene tree than the distance between the two original nodes in the species tree. d_1' and d_2' are the distances from $\Phi(u)$ and $\Phi(v)$ to their last common ancestor λ in T. Hence $d_1' + d_2'$ will be the distance from $\Phi(u)$ to $\Phi(v)$ in T. Also, $d_1 + d_2$ is the distance of the shortest path from u to v in the gene tree. Therefore, we must have $d_1 + d_2 \geq d_1' + d_2'$. In step (1), if $d_1 + d_2 < d_1' + d_2'$, then there is no possible way to map u and v to T that can achieve $D(\Phi(u), \Phi(v)) = d(u, v)$. Therefore B is not reconcilable. Step (b)(ii) is proved similarly.

For step (b)(iii), i.e. $\epsilon = 0$ and $\Phi(m) = \lambda$, since t_u, t_v, and t_z all have been uniquely reconciled based on part (A), the only way to map $\Phi(m)$ is to map to λ such that there is a three way split of the root of the reconciled gene tree with three descendant subtrees, t_u, t_v, and t_z. Hence, B is not reconcilable.

Therefore, in ReconcileTrees, if the procedure MapGeneTreeNode rejects its input then the tree B is not isometrically reconcilable. It follows that if ReconcileTrees rejects its input then it is not isometrically reconcilable. Combined with part (A), this establishes that the ReconcileTrees algorithm is correct.

(C) We prove that the algorithm runs in polynomial time.

For each node in a gene tree, the running time of MapGeneTreeNode depends on the procedure of finding the last common ancestor between two points on branches in T. The crude (non-amortized) bound on the running time for this procedure is $O(n)$, where n is the total number of tree nodes in species tree T. Therefore, the overall computational complexity of ReconcileTrees can be bounded in $O(nm)$, where m represents the total number of tree nodes in all gene trees. ■

ACKNOWLEDGMENTS

We thank Olivier Jaillon (Centre National de Sequencage, France) for providing data from the ciliate *Paramecium tetraurelia*. We also would like to thank the anonymous reviewers for critical suggestions. J.M., B.J.R., and D.H. were supported by NHGRI grant 1P41HG02371, NCI contract 22XS013A, and D.H. additionally by the Howard Hughes Medical Institute. L.Z. was supported by ARF grant 146-000-068-112. A.R. and W.M. were supported by NIH grant HG02238.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Alekseyev, M.A., and Pevzner, P.A. 2007. Whole genome duplications and contracted breakpoint graphs. *SIAM J. Comput.* 36, 1748–1763.

Aury, J., Jaillon, O., Duret, L., et al. 2006. Global trends of whole-genome duplications revealed by the ciliate *Paramecium tetraurelia*. *Nature* 444, 171–178.

Bansal, M.S., Burleigh, J.G., Eulenstein, O., et al. 2007. Heuristics for the gene-duplication problem: a $\Theta(n)$ speed-up for the local search. *RECOMB 2007* 238–252.

Boesch, F.T., and Gimpel, J.F. 1977. Covering points of a digraph with point-disjoint paths and its application to code optimization. *J. ACM* 24, 192–198.

Bourque, G., and Pevzner, P.A. 2002. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Res.* 12, 26–36.

Bourque, G., Tesler, G., and Pevzner, P.A. 2006. The convergence of cytogenetics and rearrangement-based models for ancestral genome reconstruction. *Genome Res.* 16, 311–313.

Caprara, A. 1999. Formulations and hardness of multiple sorting by reversals. *RECOMB 1999* 84–94.

Chen, K., Durand, D., and Farach-Colton, M. 2000. NOTUNG: a program for dating gene duplications and optimizing gene family trees. *J. Comput. Biol.* 7, 429–447.

Eichler, E.E., and Sankoff, D. 2003. Structural dynamics of eukaryotic chromosome evolution. *Science* 301, 793–797.

El-Mabrouk, N. 2002. Reconstructing an ancestral genome using minimum segments duplications and reversals. *J. Comput. Syst. Sci.* 65, 442–464.

El-Mabrouk, N., and Sankoff, D. 2003. The reconstruction of doubled genomes. *SIAM J. Comput.* 32, 754–792.

Fitch, W.M. 1971. Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Zool.* 20, 406–416.

Fitch, W.M. 2000. Homology, a personal view on some of the problems. *Trends Genet.* 16, 227–231.

Froenicke, L., Caldes, M.G., Graphodatsky, A., et al. 2006. Are molecular cytogenetics and bioinformatics suggesting diverging models of ancestral mammalian genomes? *Genome Res.* 16, 306–310.

Goodman, M., Czelusniak, J., Moore, G.W., et al. 1979. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst. Zool.* 28, 132–163.

Guigo, R., Muchnik, I., and Smith, T. 1996. Reconstruction of ancient molecular phylogeny. *Mol. Phylogenet. Evol.* 6, 189–213.

Ma, B., Li, M., and Zhang, L. 2000. From gene trees to species trees. *SIAM J. Comput.* 30, 729–752.

Ma, J., Zhang, L., Suh, B.B., et al. 2006. Reconstructing contiguous regions of an ancestral genome. *Genome Res.* 16, 1557–1565.

Marron, M., Swenson, K.M., and Moret, B.M.E. 2004. Genomic distances under deletions and insertions. *Theor. Comput. Sci.* 325, 347–360.

Moran, S., Newman, I., and Wolfstahl, Y. 1990. Approximation algorithms for covering a graph by vertex-disjoint paths of maximum total weight. *Networks* 20, 55–64.

Moret, B.M.E., Wyman, S.K., Bader, D.A., et al. 2001. A new implementation and detailed study of breakpoint analysis. *PSB* 583–594.

Murphy, W.J., Larkin, D.M., Everts-van der Wind, A., et al. 2005. Dynamics of mammalian chromosome evolution inferred from multispecies comparative maps. *Science* 309, 613–617.

Nadeau, J.H., and Taylor, B.A. 1984. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc. Natl. Acad. Sci. USA* 81, 814–818.

Pe'er, I., and Shamir, R. 1998. The median problems for breakpoints are NP-complete. *Electronic Colloq. Comput. Complexity* 5.

Pevzner, P., and Tesler, G. 2003. Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res.* 13, 37–45.

Rasmussen, M.D., and Kellis, M. 2007. Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes. *Genome Res.* 17, 1932–1942.

Rocchi, M., Archidiacono, N., and Stanyon, R. 2006. Ancestral genomes reconstruction: an integrated, multi-disciplinary approach is needed. *Genome Res.* 16, 1441–1444.

Sankoff, D. 1999. Genome rearrangement with gene families. *Bioinformatics* 15, 909–917.

Sankoff, D., and Blanchette, M. 1998. Multiple genome rearrangement and breakpoint phylogeny. *J. Comput. Biol.* 5, 555–570.

Sankoff, D., and El-Mabrouk, N. 2000. Duplication, rearrangement and reconciliation, 537–550. In: Sankoff, D., and Nadeau, J.H., eds. *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and the Evolution of Gene Families*. Kluwer Academic Publishers, Amsterdam.

Schwartz, S., Kent, W.J., Smit, A., et al. 2003. Human-mouse alignments with BLASTZ. *Genome Res.* 13, 103–107.

Seoighe, C., and Wolfe, K.H. 1998. Extent of genomic rearrangement after genome duplication in yeast. *Proc. Natl. Acad. Sci. USA* 95, 4447–4452.

Zhang, Y., Song, G.T., Vinar, T., et al. 2008. Reconstructing the evolutionary history of complex human gene clusters. *RECOMB 2008*, 29–49.

Zheng, C., Zhu, Q., and Sankoff, D. 2006. Genome halving with an outgroup. *Evol. Bioinform.* 2, 319–326.

Address reprint requests to:
Dr. Jian Ma
Center for Biomolecular Science and Engineering
University of California
Santa Cruz, CA 95064

E-mail: jianma@soe.ucsc.edu