A Heuristic Algorithm for Reconstructing Ancestral Gene Orders with Duplications

Jian Ma[1], Aakrosh Ratan[2], Louxin Zhang[3], Webb Miller[2], and David Haussler[1]

[1] Center for Biomolecular Science and Engineering,
University of California, Santa Cruz, CA 95064, USA
jianma@soe.ucsc.edu
[2] Center for Comparative Genomics and Bioinformatics
Penn State University, University Park, PA 16802, USA
[3] Department of Mathematics,
National University of Singapore, Singapore 117543

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 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 genomes of ciliate *Paramecium tetraurelia*.

Keywords: gene order reconstruction, duplication, contiguous ancestral region.

1 Introduction

The increasing number of genome sequences becoming available makes it feasible to computationally reconstruct ancient genomes of related species that have undergone genome rearrangements. The heart of this problem is to "undo" these large scale rearrangements and restore the ancestral gene order. Previous studies mainly focused on solving the median problem, which is either based on reversal (inversion) distance or breakpoint distance. In this problem one tries to reconstruct the common ancestor of two descendant genomes using one additional outgroup genome. Unfortunately, the median problem doesn't have exact and efficient algorithms [1,2]. In the past, heuristic programs for both breakpoint median problem and reversal median problem have been proposed [3,4,5]. But the discrepancy between the computational prediction and the result from cytogenetic experiments [6,7] suggests a need to explore further computational methods for ancestral genome reconstruction.

In our recent work [8], we proposed a new approach for reconstructing the ancestral order based on the adjacencies of orthologous genomic content in modern

G. Tesler and D. Durand (Eds.): RECOMB-CG 2007, LNBI 4751, pp. 122–135, 2007.

species, which essentially avoids solving any rearrangement median problem. The critical procedure of the method is analogous to Fitch's parsimony algorithm [9]. Instead of inferring ancestral nucleotides, we infer the locally parsimonious predecessor and successor relationships of the orthologous conserved segments in the ancestor, in this 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 [10].

However, the main drawback of the method in [8] is that it doesn't handle duplications. Indeed, duplications (including segmental duplications and tandem duplications) have a great impact on genome evolution [11]. Some previous theoretic studies [12,13,14] have included duplications (sometimes with loss) along with rearrangements. In this paper, we extend the method in [8] and propose an efficient heuristic approach to incorporate duplications into analysis when we are inferring ancestral gene orders.

2 Methods

2.1 Definitions

In this paper, we use the term **gene** to represent an atomic evolutionary unit that has never been broken due to breakpoints caused by any operations (duplication or rearrangement). If two genes are derived from a common ancestral gene, then they belong to the same **gene family**. We use $g[x]$ to represent the gene x in genome g. Also, if two genes from the same family x are in the same genome g, then we denote these genes as $g[x.i]$ and $g[x.j]$ $(i \neq j)$. 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 (−). The **reverse complement** of a chromosome is obtained by reversing the list and flipping the sign of each gene. A **genome** is a set of chromosomes.

If genome g contains gene x, then the **predecessor** $p_g(x)$ is defined as the gene that immediately precedes x on the same chromosome. Predecessor has a sign. In the opposite orientation, $p_g(-x)$ immediately precedes $-x$ in the reverse complement of the same chromosome. We set $p_g(x) = \Phi_A$ if x appears first on a chromosome. The **successor** $s_g(x)$ of x is defined analogously. And we also set $s_g(x) = \Phi_Z$ if x appears last on a chromosome. For instance, let g have the chromosome $(1\ -4.1\ -3\ 4.2\ 5\ 2)$. Then $p_g(1) = \Phi_A$, $p_g(2) = 5$, $p_g(-3) = -4.1$, $s_g(-4.1) = -3$, $p_g(-1) = 4.1$, $s_g(-5) = -4.2$, etc.

In addition to speciation events, the original ancestral genes evolve through large-scale evolutionary operations which include insertion/deletion, rearrangements (inversion, translocation, fusion/fission), and tandem and segmental duplications. Consequently, we have a different number of genes and different gene orders in present day genomes. Our goal is to reconstruct the order and orientation of genes in the target ancestral genome. We call each reconstructed chromosome a **contiguous ancestral region (CAR)**.

2.2 Species Tree, Gene Tree, and Reconciled Tree

A **species tree** is a full binary tree describing the phylogeny among different species (Fig.1(A)). All the bifurcating ancestral nodes represent speciation events, while leaves correspond to modern species. Each branch in the tree has branch length d indicating the evolutionary distance. Along the branch between two species (from ancestor to descendant), evolutionary operations could happen. In this paper, we assume that the species tree is already known, and it has been rooted and directed.

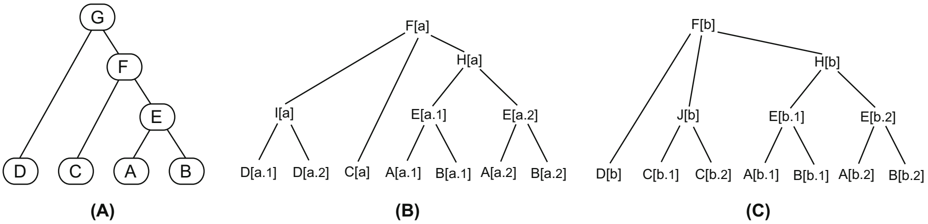

Fig. 1. (A) Species tree of modern species A, B, C, and D. Gene trees of gene family a and b are in (B) and (C), respectively. Branch length $d(D[a.1], I[a]) + d(I[a], F[a])$ in (B) is equivalent to the branch length $d(D, G) + d(F, G)$ in the species tree. We also have $d(D[b], F[b]) > d(D, G) + d(F, G)$. For other branch lengths in the gene trees, we have: $d(A[a.1], E[a.1]) = d(A[a.2], E[a.2]) = d(A[b.1], E[b.1]) = d(A[b.2], E[b.2])$, $d(B[a.1], E[a.1]) = d(B[a.2], E[a.2]) = d(B[b.1], E[b.1]) = d(B[b.2], E[b.2])$, $d(D[a.1], I[a]) = d(D[a.2], I[a])$, $d(C[b.1], J[b]) = d(C[b.2], J[b])$, $d(E[a.1], H[a]) = d(E[a.2], H[a]) = d(E[b.1], H[b]) = d(E[b.2], H[b])$, $d(H[a], F[a]) = d(H[b], F[b])$, $d(C[a], F[a]) = d(C[b.1], J[b]) + d(J[b], F[b])$.

A **gene tree**, on the other hand, is an unrooted tree, characterizing the relationships among genes in the same gene family across different species (Fig.1(B) and (C)). It also has branch lengths associated with each branch in the tree. In this paper, we have two assumptions for gene trees: (1) the duplication events have been dated and they are consistent with what happened in nature, e.g. duplication event $I[a]$ in Fig.1(B); (2) in the gene tree, all the branch lengths are exact. Therefore, if in the following reconciliation step, a node in the gene tree turns out to correspond to a speciation event, then it has a perfect match to the node in the species tree, e.g. in Fig.1 the distance from $A[a.1]$ to $E[a.1]$ in (B) is exactly the same as the distance from A to E in (A), i.e. $d(A[a.1], E[a.1]) = d(A, E)$.

A **reconciled tree** is a mapping between all gene trees and the species tree with gene duplications and losses being postulated [15]. In order to get the reconciled tree, we merge the unrooted gene trees into the rooted species tree. A reconciled tree, denoted as $\mathbb{T}$, represents all speciation and duplication events that have left a record of their effects in the leaf genomes. We start with the species tree and reconcile the gene trees into it one at a time. The species tree as well as the two gene trees in Fig.1 can be reconciled into Fig.2(A). Our reconciliation algorithm is less complicated than the traditional methods, e.g. [16] and [17], because in our case the true species tree is known and the distances in the gene trees are exact. (See Appendix for detailed reconciliation algorithm).

Each reconciliation labels the bifurcating nodes of the gene tree being reconciled as either duplication nodes or speciation nodes, maps the speciation nodes to the corresponding speciation nodes in the species tree, and maps the duplication nodes to inferred duplication nodes along the branches of the species tree (Fig.2(A)). The final reconciled tree includes these additional duplication nodes (Fig.2(B)). Each node in the reconciled tree is a genome. If there are duplications that occurred before the root of the species tree, then the root of the reconciled tree is an ancestor of the species tree root, and these ancient duplications are represented on an additional path leading from the root in reconciled tree to the root of the species tree within it, e.g. node K in Fig.2.

During the reconciliation, genes are also added along the branches of each gene tree to represent intermediate forms that are inferred to have existed at duplication branches but do not appear in the original gene tree for the family (Fig.2(A), e.g. gene a in J). The resulting gene trees are called **augmented gene trees** and denoted T_a for gene family a. For each node x in T_a, there is a **mapping** ϕ that maps x to a node y in $\mathbb{T}$, i.e. $y = \phi(x)$, indicating the genome y that gene x belongs to. Also, the root of an augmented gene tree need not always

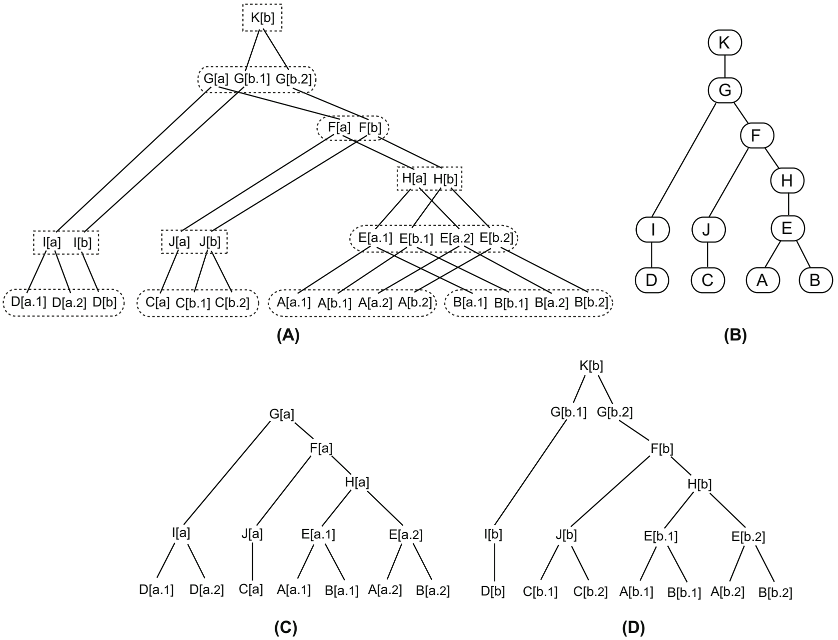

Fig. 2. (A) The reconciled tree from species tree and gene trees of gene family a and b. Node I, J, K, and H show four duplications; K is an ancient duplication. (B) A simplified form of reconciled tree $\mathbb{T}$. (C) augmented gene tree T_a. (D) T_b.

map to the root of the reconciled tree. If the gene family is first introduced by an insertion event, then the last common ancestor in the reconciled tree of all the observed genes in the family may be a node below the root. For example, in T_a the root does not map to K in the reconciled tree. We could interpret gene family a as an insertion before G but after K in the reconciled tree.

Along branch h to g in the reconciled tree, we define $\tilde{A}_h(g[x])$ as the **direct ancestor** of $g[x]$ in h and $\tilde{D}_g(h[x])$ as the set of **direct descendants** of $h[x]$ in g. Note that $\tilde{D}(h[x])$ could contain two descendants if x is duplicated at h. If $g[x]$ has no ancestor, then $\tilde{A}_h(g[x]) = \emptyset$. Conversely, if $h[x]$ has no descendant in g, $\tilde{D}_g(h[x]) = \emptyset$. For example, in Fig.2, $\tilde{A}_E(B[a.1]) = E[a.1]$, $\tilde{D}_C(J[b]) = \{C[b.1], C[b.2]\}$.

2.3 Reconstructing Ancestral Adjacency

After obtaining a reconciled tree $\mathbb{T}$ and augmented gene trees T_i (for all gene family i), our goal is to determine a set of lists of gene orders that closely approximates the genome structure of the species corresponding to a target ancestral genome in $\mathbb{T}$.

For any genome g, we associate with each gene x two sets of signed genes, denoted $P_g(x)$ and $S_g(x)$, giving potential predecessors and successors of x relative to chromosomes of g. If g is a modern genome, $P_g(x) = \{p_g(x)\}$ and $S_g(x) = \{s_g(x)\}$, for each x. If g does not contain x, then both sets are empty. We also define that $\tilde{A}_h(P_g(x)) = \{\tilde{A}_h(y_i) \mid y_i \in P_g(x)\}$. $\tilde{D}_h(P_g(x))$ can be defined analogously.

We use N_g to denote the number of genes in genome g, which can be counted directly from the reconciled tree. For example, $N_E = 4$ in the example in Fig.2.

The inference procedures of predecessor and successor associated with each gene in the gene tree is similar to the method in [8]. The first stage of the algorithm works in a bottom-up fashion. The general idea is that, for each node π in the gene tree, we compute its predecessor set according to the following rule: If π is a leaf, then predecessor set consists of the unique predecessor. Otherwise, assume π has children τ and φ; then, the predecessor set is equal to the intersection or union of the predecessor sets of τ and φ depending on whether their predecessor sets are disjoint or not. The second stage works in a top-down fashion to adjust the predecessor sets. Similarly, we infer the successors.

The procedure Get-Predecessor-Successor-Bottom-Up($root(\mathsf{T}_i)$) constructs $P_g(x)$ and $S_g(x)$ for each gene x of gene family i in every ancestral genome g, where $root(\mathsf{T}_i)$ denotes the root of T_i. Suppose π is the current node and τ and φ are the two direct descendants of π in T_i. Note that either τ or φ might be null.

```
Get-Predecessor-Successor-Bottom-Up($\pi$)
if $\pi$ is not null and $\pi$ is non-leaf node
   then Get-Predecessor-Successor-Bottom-Up($\tau$)
        Get-Predecessor-Successor-Bottom-Up($\varphi$)
        $h \leftarrow \phi(\pi)$; $f \leftarrow \phi(\tau)$; $g \leftarrow \phi(\varphi)$
        if $\| \tilde{A}_h(P_f(\tau)) \cap \tilde{A}_h(P_g(\varphi)) \| \neq 0$
```

```
            then $P_h(\pi) \leftarrow \tilde{A}_h(P_f(\tau)) \cap \tilde{A}_h(P_g(\varphi))$
            else $P_h(\pi) \leftarrow \tilde{A}_h(P_f(\tau)) \cup \tilde{A}_h(P_g(\varphi))$
        if $\| \tilde{A}_h(S_f(\tau)) \cap \tilde{A}_h(S_g(\varphi)) \| \neq 0$
            then $S_h(\pi) \leftarrow \tilde{A}_h(S_f(\tau)) \cap \tilde{A}_h(S_g(\varphi))$
            else $S_h(\pi) \leftarrow \tilde{A}_h(S_f(\tau)) \cup \tilde{A}_h(S_g(\varphi))$
```

The root of the reconciled tree $\mathbb{T}$ is not always the target genome we want to reconstruct. Therefore, we first infer $P_R(x)$ and $S_R(x)$ in the common ancestor R in $\mathbb{T}$. Then we propagate $P_R(i)$ and $S_R(i)$ down the tree until we reach the target ancestor α. We use Adjust-Ancestor-Top-Down to adjust the original $P_g(x_i)$ and $S_g(x_i)$ for every gene x_i in genome g leading from R to α, assuming that the path from R to α has already been recorded (the .next field means the next node on the path from R to α).

```
Adjust-Ancestor-Top-Down($R, \alpha$)
$h \leftarrow R$; $g \leftarrow R.next$
while $h \neq \alpha$
    do for each $x_i \in \mathbf{X}$ where $\mathbf{X} = x_1, -x_1, ..., x_{N_g}, -x_{N_g}$
            do if $\| \tilde{A}_h(P_g(x_i)) \cap P_h(\tilde{A}_h(x_i)) \| \neq 0$
                    then $P_g(x_i) \leftarrow P_g(x_i) \cap \tilde{D}_g(\tilde{A}_h(P_g(x_i)) \cap P_h(\tilde{A}_h(x_i)))$
                if $\| \tilde{A}_h(S_g(x_i)) \cap S_h(\tilde{A}_h(x_i)) \| \neq 0$
                    then $S_g(x_i) \leftarrow S_g(x_i) \cap \tilde{D}_g(\tilde{A}_h(S_g(x_i)) \cap S_h(\tilde{A}_h(x_i)))$
        $h \leftarrow g$; $g \leftarrow g.next$
```

At this point, in the target ancestor α, we have had potential predecessors and successors for each gene. The remaining task is to reconstruct the order based on adjacency information.

2.4 From Ancestral Adjacency to Ancestral Gene Order

We first construct a **predecessor graph** G^P_α and a **successor graph** G^S_α for the target genome α. The digraph $G^P_\alpha = (V, E)$, where $|V| = 2N_\alpha$, is defined such that each genex_i corresponds to two nodes, i and $-i$, and the set of directed edges is: $E(G^P_\alpha) = \{(u, v) \mid u \in P_\alpha(v)\}$. Similarly, in digraph $G^S_\alpha = (V, E)$, $|V| = 2N_\alpha$, and: $E(G^S_\alpha) = \{(u, v) \mid v \in S_\alpha(u)\}$. Here, (u, v) denotes an arc directed from u to v. Note that an edge in G^P_α is *from* the predecessor, while an edge in G^S_α is *to* the successor. For instance, let g have the chromosome (1 -4 -3 5.1 2). Then G^P_g and G^S_g are as shown in Fig.3(A) and (B), respectively.

We intersect G^P_α and G^S_α, producing the intersection graph $G = G^P_\alpha \cap G^S_\alpha$, retaining edges that are not connecting to either of the endpoints, Φ_A and Φ_Z. Then special care is taken to add endpoint edges, basically retaining all the endpoint edges that appear in both G^P_α and G^S_α. All three graphs (predecessor, successor, and intersection) have the same set of $2N_\alpha$ nodes. G's edges are:

$$E(G) = \{E(G^P_\alpha) \cap E(G^S_\alpha)\} \cup \{(\Phi_A, v) \mid (\Phi_A, v) \in E(G^P_\alpha)\} \cup \{(u, \Phi_Z) \mid (u, \Phi_Z) \in E(G^S_\alpha)\} \quad (1)$$

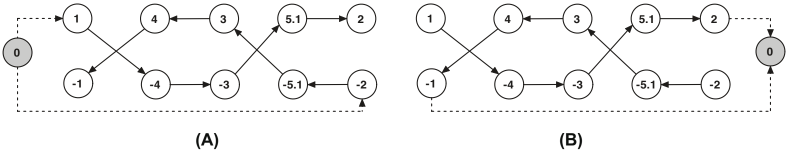

Fig. 3. (A) A predecessor graph G_g^P; (B) A successor graph G_g^S

The edges of the intersection graph G indicate consistent predecessor and successor relationships that are supported by $\mathbb{T}$, T_i and the modern genomes. However, they do not necessarily indicate a unique adjacency relationship for a particular gene. Three potential ambiguous cases might occur in the intersection graph, 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)).

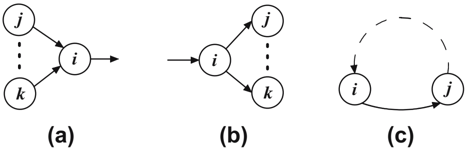

Fig. 4. Three potential ambiguous cases in the intersection graph G

If none of these ambiguous cases is present, the intersection graph 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 the ambiguity and choose appropriate directed edges to form CARs. We assign a weight to each of the directed edges in the remaining graph using the following approach.

For an directed edge (i, j), if $outdegree(i) = 1$ and $indegree(j) = 1$ (in other words, it is not among one of the incoming edges of case (a) nor it is among one of the outgoing edges of case (b)), we set $w_\alpha(i, j) = 1$. Otherwise, the corresponding weight $w_\alpha(i, j)$ is determined recursively by:

$$w_\alpha(i,j) = \frac{d(\alpha,\tau) \cdot w_\varphi(i,j) + d(\alpha,\varphi) \cdot w_\tau(i,j)}{d(\alpha,\tau) + d(\alpha,\varphi)} \tag{2}$$

where $d(\alpha, \tau)$ and $d(\alpha, \varphi)$ are the branch lengths to the left child and right child; $w_\tau(i, j)$ and $w_\varphi(i, j)$ are the edge weights on left child and right child, respectively. On a leaf genome, if (i, j) is present in the predecessor graph, we set $w(i, j) = 1$, otherwise $w(i, j) = 0$. This kind of edge weight can also be determined by a postorder traversal. Note that if an edge (i, j) is involved in ambiguous case (a) or (b), $w(i, j) < 1$. The underlying assumption of equation 2 is that rearrangement is more likely to happen on longer branches.

Our goal is to connect elements into the longest possible CARs that are consistent with the observed data. The problem can be transformed into looking for vertex-disjoint paths that cover all the nodes in the digraph G with the maximum weight. Here we also allow degenerate paths, where there is only one node. The simplified version of this problem when all the edge weights are the same, say 1, is equivalent to the Minimum Path Cover Problem, i.e., finding the minimum number of vertex-disjoint paths covering all the nodes in the digraph. The minimum path cover problem was proved to be NP-hard [18].

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

```
Find-Cars($G$)
1  Sort edges by weight in descending order.
2  Create a new graph $C$, $V(C) = V(G)$ and $E(C) = \emptyset$
3  for each available $(i, j) \in E(G)$, in order of edge weight
4     do if $outdegree(i) = 0$ and $indegree(j) = 0$
5           then Add edge $(i, j)$ and $(-j, -i)$ to $E(C)$
6                Update $outdegree(i)$ and $indegree(j)$ in $C$
7  Break cycles in $C$.
8  return $C$.
```

Note that the simple greedy process doesn't guarantee there will be no cycle in the path set. We need a final step (line 7) to detect and break the cycles. We use the depth-first-search algorithm to detect cycles in graph G. In fact, we can prove that if there is a cycle, the weight of each edge in that cycle is 1. Therefore, we can simply discard an arbitrary edge to break the cycle (In a variant where circular chromosomes are considered, then cycles would be allowed). The remaining paths in G correspond to the CARs we want to reconstruct.

When adding edges into an existing path, particular care is needed to avoid putting j and $-j$ in the same CAR. In addition, we 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.

2.5 Summary

In outline, the whole Infer-Cars-With-Dup algorithm can be described as follows, where α is the target ancestor, and $\mathbb{G}$ denotes the collection of modern genomes.

```
INFER-CARS-WITH-DUP($\alpha$)
Construct $\mathbb{T}$ and $\mathsf{T}_x$ (for each gene family $x$)
$\mathbb{C} \leftarrow$ empty set of CARs
$R \leftarrow root(\mathbb{T})$
Initialize $P_g(i)$ and $S_g(i)$ for each gene $i$ in every $g$ in $\mathbb{G}$
for each gene family $i$
    do GET-PREDECESSOR-SUCCESSOR-BOTTOM-UP($root(\mathsf{T}_i)$)
ADJUST-ANCESTOR-TOP-DOWN($R, \alpha$)
Get graph $G$ according to Equation (1)
$\mathbb{C} \leftarrow$FIND-CARS($G$)
return $\mathbb{C}$
```

3 Results

3.1 Simulation Results

We used extensive simulations to test and validate our analysis. The simulator starts with a hypothetical 'ancestor' genome which evolves into the extant species through speciation, inversion, translocation, fusion, fission, insertion, deletion, and duplication. When an operation is applied, the breakpoint is chosen uniformly at random from the set of used or unused breakpoints on this chromosome, depending on the breakpoint reuse ratio. The length of the operation is also picked uniformly at random within the specified distance from the first breakpoint.

We tuned the weights of these operations in order to generate simulated data that makes more biological sense specifically for placental mammalian genomes. The ancestor genome was assigned around 5,000 genes. The parameters or weights of the large scale operations were tuned such that the extant species had around the same number of genes. The breakpoint reuse ratio was kept around 8%-10% and each of the extant species had 5%–10% duplicated genes. We simulated 50 datasets using the phylogenetic tree:

`((((human,chimp),rhesus),(mouse,rat)),dog)`.

On average, the ratio of breakpoint reuse is 9.98%, the ratio of duplicated genes in each extant species is 8.12% (rhesus), 7.52% (human), 7.26% (chimp), 7.12% (mouse), 7.85% (rat), and 7.23% (dog), respectively. Also, rearrangements are distributed as 82.33% inversions, 9.40% translocations, 3.86% fusions, and 4.40% fissions. In all the duplication events, 30.40% are tandem duplications and 69.60% are segmental duplications.

We ran our reconstruction program for inferring CARs on each dataset (avg. running time 14.62min) and compared the predicted adjacencies with the known (simulated) ones. Our target ancestor was primate-rodent ancestor and dog was treated as outgroup. For determining the success rate, we considered only the effective ancestral adjacencies (˜59% of all ancestral adjacencies) that were broken in at least one lineage in the subtree rooted by primate-rodent ancestor, since the unbroken adjacencies will be found by essentially any procedure.

The frequency of correctly predicted adjacencies was 99.46% (SD=0.43%) for the primate-rodent ancestor. The reconstruction accuracy of human-rhesus ancestor and mouse-rat ancestor is 99.75% (SD=0.27%) and 99.72% (SD=0.25%) respectively.

We did some additional experiments to see how the performance changes in the primate-rodent ancestor if we change parameters in the simulation. We made the effective ancestral adjacency vary by using different number of rearrangement operations. Interestingly, the accuracy didn't change much. For example, when the effective ancestral adjacency is around 10%, the accuracy is 99.67%. When the effective adjacency is around 70%, the accuracy is 99.45%. We think the accuracy didn't really depend on effective adjacency because we used six species in this simulation. We also increased the breakpoint reuse ratio to around 40% when the effective adjacency ratio is 70%, then the accuracy dropped to 96.83%. We concluded from these preliminary experiments that when the number of leaf genomes is reasonable, the reconstruction performance isn't hurt much if we increase the number of operations (as reflected in the effective adjacencies). Instead, the performance will be suffered if we increase the breakpoint reuse ratio to let one ancestral adjacency be broken independently in different lineages.

3.2 Application to Real Data

It has been shown that the unicellular eukaryote *Paramecium tetraurelia*, a ciliate, which contains about 40,000 genes, is a result of at least three whole genome duplication (WGD) with additional rearrangement operations [19]. In that paper, the 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 deriving from a common ancestral region. Then paralogons were merged into single ancestral blocks and the process was iterated until reaching the ancient WGD. However, they didn't intend to figure out the gene orders in each ancestral block. When a paralogon was constructed, the detailed order and orientation of genes inside the block were ignored.

All 39,642 genes form 22,635 gene families (including 11,740 single-gene families), which have been scattered on 676 scaffolds in the present day genome. 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 [19]. The reconciled tree contains one leaf genome, which is the modern genome, as well as ancestral nodes representing duplication events. We built the augmented gene trees accordingly.

Many genes do not have paralogous genes in the paralogons for a particular ancestral genome. If we 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. For example, if we include all the genes, there are 1,937 reconstructed CARs in the old WGD. Therefore, when we were reconstructing CARs in a certain target genome, we did some

Table 1. Number of CARs we reconstructed in three target ancestral genomes

target ancestor	genes we included	anc genes with paralog (from [19])	gene families we used	predicted CARs	anc blocks in [19]
Old WGD	2,981	1,530	559	57	43
Intermediary WGD	11,620	7,996	3,770	144	81
Recent WGD	25,708	24,052	9,951	228	131

preprocessing to only retain genes that have paralogous genes derived from more ancient duplications. Additional genes were also 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 [19] using the paralogon method to construct ancestral blocks. There are two reasons for this: (1) The authors of [19] ignored the gene orders while we take order andorientation 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.

Since paper [19] didn't reconstruct the ancestral gene adjacencies, we couldn't compare our prediction with theirs in detail. Preliminary comparison showed that our prediction has basically and more detailed refinement than the result from [19]. Also, recent studies on genome halving problem [20,21,22,23] might be particularly useful and interesting to be applied to the Paramecium genome. As further ciliate genomes become available, we plan to further investigate the changes of gene orders between different WGDs, using additional outgroup information from closely related species to pick up more adjacencies we couldn't reconstruct now, which will help to determine which methods of reconstructing ancestral architecture are best, and might shed more light on the evolution of the ciliate *Paramecium tetraurelia*.

4 Discussion

In this paper, we extend the method in [8] to reconstruct ancestral gene orders with duplications. We have a simplifying assumption 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 gene trees with approximate distances. Therefore, a more robust reconciliation method is needed, e.g. [24] and [25]. 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 the descendant genome and the ancestor genome have been both predicted.

Our simulation on large-scale mammalian genome evolution looks promising. However, 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 and accurately dating the duplication events.

Acknowledgement

We thank Olivier Jaillon at Centre National de Sequencage in France for providing data of ciliate *Paramecium tetraurelia*.

References

1. Caprara, A.: Formulations and hardness of multiple sorting by reversals. RECOMB, 84–94 (1999)
2. Pe'er, I., Shamir, R.: The median problems for breakpoints are NP-complete. Electronic Colloquium on Computational Complexity (ECCC), 5(71) (1998)
3. Sankoff, D., Blanchette, M.: Multiple genome rearrangement and breakpoint phylogeny. J. Comput. Biol. 5(3), 555–570 (1998)
4. Moret, B.M.E., Wyman, S.K., Bader, D.A., Warnow, T., Yan, M.: A new implmentation and detailed study of breakpoint analysis. PSB, 583–594 (2001)
5. Bourque, G., Pevzner, P.A.: Genome-scale evolution: reconstructing gene orders in the ancestral species. Genome Res. 12(1), 26–36 (2002)
6. Froenicke, L., Caldes, M.G., Graphodatsky, A., Muller, S., Lyons, L.A., Robinson, T.J., Volleth, M., Yang, F., Wienberg, J.: Are molecular cytogenetics and bioinformatics suggesting diverging models of ancestral mammalian genomes? Genome Res. Genome Res. 16(3), 306–310 (2006)
7. Bourque, G., Tesler, G., Pevzner, P.A.: The convergence of cytogenetics and rearrangement-based models for ancestral genome reconstruction. Genome Res. 16(3), 311–313 (2006)
8. Ma, J., Zhang, L., Suh, B.B., Raney, B.J., Burhans, R.C., Kent, W.J., Blanchette, M., Haussler, D., Miller, W.: Reconstructing contiguous regions of an ancestral genome. Genome Res. 16(12), 1557–1565 (2006)
9. Fitch, W.M.: Toward defining the course of evolution: minimum change for a specific tree topology. Syst. Zool. 20, 406–416 (1971)
10. Rocchi, M., Archidiacono, N., Stanyon, R.: Ancestral genomes reconstruction: An integrated, multi-disciplinary approach is needed. Genome Res. 16(12), 1441–1444 (2006)
11. Eichler, E.E., Sankoff, D.: Structural dynamics of eukaryotic chromosome evolution. Science 301(5634), 793–797 (2003)
12. Sankoff, D.: Genome rearrangement with gene families. Bioinformatics 15(11), 909–917 (1999)
13. Sankoff, D., El-Mabrouk, N.: Duplication, rearrangement and reconciliation. In: Sankoff, D., Nadeau, J.H. (eds.) Comparative genomics: Empirical and analytical approaches to gene order dynamics, map alignment and the evolution of gene families, pp. 537–550. Kluwer Academic Publishers, Dordrecht (2000)

14. Marron, M., Swenson, K.M., Moret, B.M.E.: Genomic distances under deletions and insertions. Theor. Comput. Sci. 325(3), 347–360 (2004)
15. Page, R.D.M., Charleston, M.A.: From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. Mol. Phylogenet. Evol. 7(2), 231–240 (1997)
16. Goodman, M., Czelusniak, J., Moore, G.W., Romero-Herrera, A.E., Matsuda, G.: Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from Globin Sequences. Syst. Zool. 28(2), 132–163 (1979)
17. Guigo, R., Muchnik, I., Smith, T.F.: Reconstruction of ancient molecular phylogeny. Mol. Phylogenet. Evol. 6(2), 189–213 (1996)
18. Boesch, F.T., Gimpel, J.F.: Covering points of a digraph with point-disjoint paths and its application to code optimization. J. ACM. 24(2), 192–198 (1977)
19. Aury, J.M., Jaillon, O., Duret, L., Noel, B., Jubin, C., Porcel, B.M., Ségurens, B., Daubin, V., Anthouard, V., Aiach, N., et al.: Global trends of whole-genome duplications revealed by the ciliate Paramecium tetraurelia. Nature 444, 171–178 (2006)
20. Seoighe, C., Wolfe, K.H.: Extent of genomic rearrangement after genome duplication in yeast. PNAS 95(8), 4447–4452 (1998)
21. El-Mabrouk, N., Sankoff, D.: The reconstruction of doubled genomes. SIAM J. Comput. 32(3), 754–792 (2003)
22. Alekseyev, M.A., Pevzner, P.A.: Whole genome duplications and contracted breakpoint graphs. SIAM J. Comput. 36(6), 1748–1763 (2007)
23. Zheng, C., Zhu, Q., Sankoff, D.: Genome halving with an outgroup. Evolutionary Bioinformatics 2, 319–326 (2006)
24. Chen, K., Durand, D., Farach-Colton, M.: NOTUNG: a program for dating gene duplications and optimizing gene family trees. J. Comput. Biol. 7(3-4), 429–447 (2000)
25. Bansal, M.S., Burleigh, J.G., Eulenstein, O., Wehe, A.: Heuristics for the gene-duplication problem: A $\Theta(n)$ speed-up for the local search. RECOMB, pp. 238–252 (2007)

Appendix

We discuss in detail the algorithm for determining the reconciled tree and augmented gene tree. Let S be a rooted species tree and A be an unrooted gene tree. We assume that S has an infinitely long incoming edge leading into its root to accommodate ancient duplications, if needed. A **reconciliation** of A with S is a mapping ϕ from the nodes of A into the set of nodes and points along the edges of S with the following properties: (1) Every leaf l of A maps to a leaf $\phi(l)$ of S of the same species; (2) Each internal node a of A maps to a point $\phi(a)$ in S that lies either at a node or at a point on an edge in S; and (3) The mapping ϕ is isometric in the sense that for every leaf node l in A, the distance from a to l in A is the same as the distance from $\phi(a)$ to $\phi(l)$ in S. When $\phi(a)$ is a node in the species tree S, we say that a is a *speciation node* in A, and when $\phi(a)$ is a point that lies along an edge in S we say that a is a *duplication node* in A and we create a corresponding duplication node at $\phi(a)$ in S.

Any internal node x in the unrooted binary tree A will be connected to three other nodes u, v, and w, defining three possible rooted and directed subtrees U,

V, and W of A, respectively. If A is to be successfully reconciled with S, two of these subtrees, say U and V, must map to directed subtrees of S in such a way that $\phi(x)$ lies above $\phi(u)$ and $\phi(v)$. To define the complete reconciliation, we proceed inductively, assuming that we have already reconciled subtrees U and V of A, and extending this reconciliation to include x. Let d_1' and d_2' be the distances in A from x to u and v to x, respectively. Let $\tilde{x}$ be the last common ancestor of $\phi(u)$ and $\phi(v)$ in S. Let d_1 and d_2 be the distances in S from $\phi(u)$ and $\phi(v)$ to $\tilde{x}$, respectively. We will have $d = d_1' + d_2' - d_1 - d_2 \geq 0$. Then the subtree of A rooted at x and containing U and V can be reconciled with S by extending the reconciliation of its subtrees U and V by adding a point $\phi(x)$. The point $\phi(x)$ must lie at a distance $d/2$ upstream from $\tilde{x}$ in S, along the unique path in S leading into $\tilde{x}$. Such a point always exists in S because we have added an infinitely long stem branch leading into the original root of S. If $d = 0$, then $\phi(x) = \tilde{x}$. It is clear that the distance from $\phi(x)$ to $\phi(l)$ for any leaf l in U or V must be correct, since the distances from $\phi(u)$ and $\phi(v)$ are correct by the inductive hypothesis, and the additional distance added from $\phi(u)$ or $\phi(v)$ to $\phi(x)$ in the above construction is exactly the increment needed to keep the distances correct.

So long as A has more than one node, the inductive construction terminates with two adjacent nodes y and z that dominate all other nodes in A, in the sense that both the subtree Y rooted at y and pointing away from z, and the subtree Z rooted at z and pointing away from y, are reconciled into subtrees of S. Then the final step is to determine the root of the gene tree. Now let d be the distance between y and z in A. Let $\tilde{r}$ be the last common ancestor of $\phi(y)$ and $\phi(z)$ in S. Let d_1 and d_2 be the distances in S from $\phi(y)$ and $\phi(z)$ to $\tilde{r}$, respectively. We define the root r of the gene tree A as the point at distance $(d + d_1 - d_2)/2$ from y and $(d + d_2 - d_1)/2$ from z along the edge connecting y and z, and $\phi(r)$ as the corresponding point at distance $(d - d_1 - d_2)/2$ upstream from $\tilde{r}$ in S. This completes the reconciliation.

We construct the reconciled tree by repeating the above procedure for each gene tree. Each reconciliation adds new duplication nodes to S until the final reconciled tree $\mathbb{T}$ is built.