JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 6, 2010

Pp. 767–781
DOI: 10.1089/cmb.2009.0249

Research Articles

The Imperfect Ancestral Recombination Graph Reconstruction Problem: Upper Bounds for Recombination and Homoplasy

FUMEI LAM, RYAN TARPINE, and SORIN ISTRAIL

ABSTRACT

One of the central problems in computational biology is the reconstruction of evolutionary histories. While models incorporating recombination and homoplasy have been studied separately, a missing component in the theory is a robust and flexible unifying model which incorporates both of these major biological events shaping genetic diversity. In this article, we introduce the first such unifying model and develop algorithms to find the optimal ancestral recombination graph incorporating recombinations and homoplasy events. The power of our framework is the connection between our formulation and the *Directed Steiner Arborescence Problem* in combinatorial optimization. We implement linear programming techniques as well as heuristics for the Directed Steiner Arborescence Problem, and use our methods to construct evolutionary histories for both simulated and real data sets.

Key words: recombination, phylogeny, homoplasy, genetic variation, haplotypes, sequence analysis.

1. INTRODUCTION

One of the central problems in computational biology is the problem of reconstructing evolutionary histories. Many variants of the problem have been studied, but with the growing repositories of variation data, there is increased demand for new tools for analysis. Prior studies have established that in order to accurately represent complete evolutionary histories, the underlying model must incorporate hybridization events, which correspond to the mixing of genetic material of ancestral sequences passed to their descendents. Nordborg (2001) states, ''In the era of genomic polymorphism data, the importance of modeling recombination can hardly be overemphasized.'' Another important set of events to consider in the construction of evolutionary histories are homoplasy events, which extend beyond the infinite sites model to include back and recurrent mutations.

The goal of our work is to create a model for constructing evolutionary histories that extends current models by incorporating both evolutionary processes of recombination and homoplasy. Such analyses are applicable to other research, including the study of linkage disequilibrium and recombination hotspots, and the search for genetic predictors of disease. The mathematical and computational challenge is to develop methods that are robust and rigorous.

Department of Computer Science, Brown Univerisity, Providence, Rhode Island.

2. SINGLE EVENT MODELS

In the phylogenetic tree reconstruction models under consideration, the input is a matrix with rows representing individuals (e.g., haplotype data) and columns representing sites. Throughout this article, we will assume that the input is binary, with values 0 and 1. Important examples of such instances arise from single nucleotide polymorphism (SNP) data; a SNP is a position in the genome with at least two different bases present in the population, each with a frequency above a certain threshold. It is the most abundant type of polymorphism and in practice is found to be largely binary. A vast amount of such data has been collected in the International HapMap project of the International HapMap Consortium (2005).

The two important evolutionary forces we consider in this work are mutation and recombination. A mutation is a change in a single site of a sequence, either from value 0 to 1 or from 1 to 0. In population genetics, the *infinite sites* model assumes that at most one mutation occured at each site throughout evolutionary history. A *phylogenetic tree* is a directed tree with each edge labeled by an integer between 1 and m and each node labeled by a binary sequence. If edge $e = (u, v)$ is labeled by site i, then the sequence v can be obtained from the sequence u by changing the value at site i of u from value x to value $1 - x$. We denote this by $v = u.i$. A phylogenetic tree T displaying input I satisfies the property that each row in I labels a node in tree T. In what follows, we interchangeably refer to a node in the ancestral recombination graph (ARG) and the binary sequence labeling it. If binary input I can be displayed in a phylogenetic tree such that each label from $1, 2, \ldots m$ labels at most one edge, the resulting tree is called a *perfect phylogeny*.

The other event we consider is meiotic, or crossover recombination, which is one of the dominant forces impacting genetic diversity. A crossover recombination occurs when two chromosomes of equal length exchange material to form a descendent, which contains a prefix of the first chromosome and a suffix of the second. In this work, the term recombination refers to such events.

It is well known that a set of equal length binary sequences I can be displayed in a perfect phylogeny if and only if it passes the *four gamete test* (Gusfield, 1991, 1999). Much of the current haplotype data fails the perfect phylogeny test and thus cannot be explained by a perfect phylogeny. In the following sections, we discuss two previously studied models for reconstructing evolutionary histories for such data: *ancestral recombination graphs* and *imperfect phylogenetic trees*.

2.1. Ancestral recombination graphs

An *ancestral recombination graph* or *phylogenetic network* is a directed acyclic graph in which nodes correspond to binary sequences of length m, and edges correspond to either mutation or recombination events. There is a unique root vertex with indegree 0, and every node other than the root has indegree either one or two. Nodes with indegree two are *recombination nodes*; if e is an edge that is directed into a recombination node, then e is a *recombination edge*. If s is a recombination node, then it can be obtained by its two parents by combining a prefix of one parent with a suffix of the second parent. Otherwise, e is a *mutation edge* and is labeled by a site $i \in \{1, 2, \ldots m\}$; for a mutation edge (u, v) labeled by site i, the sequences satisfy $v = u.i$. A homoplasy event occurs if there exists a site i with two or more edges labeled by i. Homoplasy events are either recurrent mutations ($x \in \{0, 1\}$ mutates to $1 - x$ two or more times at a site i) or back mutations (in which site i mutates from $x \in \{0, 1\}$ to $1 - x$ and then from $1 - x$ back to x). An ARG G displays input I if there is a node in G corresponding to each row in I. The following problem has been the subject of intensive research (Bordewich and Semple, 2007; Griffiths and Marjoram, 1997; Hein, 1990, 1993; Hudson and Kaplan, 1985; Lyngs et al., 2005; Meyers and Griffiths, 2003; Song et al., 2005; Song, 2006; Song and Hein, 2004; Wang et al., 2001).

Ancestral Recombination Graph (ARG) Reconstruction Problem: Given a set of n binary sequences I each of length m, find the ancestral recombination graph displaying I with the minimum number of recombination nodes $R_{min}(I)$ under the infinite sites model.

This problem was first considered by Hudson and Kaplan (1985). As this problem is APX-hard (and therefore NP-hard) in the general case (Bordewich and Semple, 2007), there has been increased focus to develop efficient methods to compute lower bounds for $R_{min}(I)$. Meyers and Griffiths (2003) develop methods to obtain global lower bounds by combining local lower bounds. Song and Hein (2004) introduce lower bounding methods based on set theoretic conditions and use tree operations to generate optimal ARGs. Song et al. (2005) compute both lower and upper bounds on the minimum number of recombinations needed to construct the evolutionary history. Their lower and upper bounds are shown to often

coincide in practice, and in such cases, their algorithm solves the parsimonious ARG reconstruction problem to optimality.

2.2. Imperfect phylogenetic trees

The second method to address sequences that cannot be displayed on a perfect phylogeny is the *imperfect phylogeny* reconstruction method. An imperfect phylogeny is a phylogenetic tree that violates infinite sites by allowing back and recurrent mutations. The following problem has also been the subject of a vast literature (Agarwala and Fernandez-Baca, 1994; Bandelt, 1991; Damaschke, 2004; Ganapathy et al., 2003; Semple and Steel, 2003; Sridhar et al., 2006, 2007b).

Imperfect Phylogeny Reconstruction Problem: Given a set of n binary sequences I each of length m, find a phylogenetic tree explaining I with the minimum number of homoplasy events.

Because the problem is NP-hard in general (Foulds and Graham, 1982), an important problem is to isolate parameters that capture the complexity of the problem in the hope of finding an algorithm that is polynomial in the remaining parameters. In Blelloch et al. (2006) and Sridhar et al. (2007a), it was shown that for binary sequences, the imperfect phylogeny problem can be solved in fixed parameter tractable time in parameter q, where q denotes the number of homoplasy events needed to explain the input sequences.

3. HIERARCHY OF IMPERFECT ANCESTRAL RECOMBINATION GRAPHS

While models for recombination and imperfection have been studied separately, a missing component in the theory of constructing evolutionary histories is a robust unifying model incorporating both phenomena.

The following characteristics should be satisfied by any model incorporating both recombination and homoplasy events.

1. Robustness: incorporates both single-event models (ARG reconstruction and imperfect phylogeny reconstruction) as special cases
2. Flexibility: model input incorporates a weighted set of parameters based on information about the relative rates of recurrent mutation and recombination events for the input sequences.
3. Computational effectiveness: allows algorithms on real data

An *imperfect ancestral recombination graph* on input I is an ARG displaying I which allows homoplasy events. To satisfy Property (2), we would like to be able to input cost parameters associated to the mutation and recombination events obtained from separate analysis. The *weighted cost* of an imperfect ARG A is the sum of the costs associated to the recombination and mutation events occuring in A. The imperfect ARG reconstruction problem is the following.

Imperfect Ancestral Recombination Graph (Imperfect ARG) Reconstruction Problem: Given a set of n binary sequences I each of length m, a common ancestral sequence r, and weights w, find an Imperfect ARG displaying I with minimum weighted cost.

This is an important problem, as it combines the two major biological events shaping genetic diversity into a single framework. By choosing a sufficiently large value for the cost of recombination, the problem includes the imperfect phylogeny reconstruction problem as a special case. Similarly, by choosing a sufficiently large value for the cost of mutation, the problem includes the ARG reconstruction problem as a special case. It follows that the imperfect ARG reconstruction problem is also APX-hard (and therefore NP-hard) in the general case.

Our contribution is to develop algorithms for the imperfect ARG reconstruction problem. Moreover, our model has the advantage that it uses the same representation for both recombination and mutation events, rather than using cycles to represent recombination and edges to represent mutation. We believe this uniform treatment of the two events is advantageous in the development of algorithms.

The central idea for the construction of the graph theoretic representations we will describe is the transformation of recombination cycles into simpler graph structures. Such a transformation is powerful because, with cycles no longer present, the problem of constructing a minimum ARG can be formulated as a well-known combinatorial optimization problem known as the Minimum Directed Steiner Arborescence (MDSA) problem (for a survey and applications of the MDSA problem, see Hwang et al., 1992, and Winter, 1987).

3.1. Directed Steiner arborescence problem

Given a connected directed graph G with edge weights w_e, a root vertex r, and a set of terminal vertices V_T, a *directed Steiner arborescence* is a subgraph of G that contains a directed path from r to each terminal in V_T. The *cost* of a directed Steiner arborescence D is the sum of its edge weights. The following is a well-studied problem in combinatorial optimization.

Minimum Directed Steiner Arborescence (MDSA) Problem

Input: Connected directed graph G with edge weights w_e, root vertex r and a set of terminal vertices V_T
Objective: Find a directed Steiner arborescence of minimum weight in G.

The MDSA problem is NP-hard (Karp, 1972) and as hard to approximate as the Set Cover Problem (Guha and Khuller, 1996). Therefore, there is no constant factor approximation algorithm for the problem (unless $\mathrm{P} = \mathrm{NP}$).

3.2. Informal description of main result

We begin by giving an informal outline of our method and main results. Throughout the discussion, the input is assumed to be a set of n binary sequences, each of length m. For input I, consider first the set $\mathcal{A}$ of all ancestral recombination graphs displaying I. We will construct a set of auxiliary graphs, called *hierarchy graphs* (and denoted $\mathcal{HG}$), together with a sequence of transformations that maps each ancestral recombination graph to a hierarchy graph. To informally describe one level of hierarchy graphs, we will first utilize two pebbles (one labeled a and one labeled b) and describe a sequence of moves for these pebbles. We begin by detailing how the transformation turns a single recombination cycle into a directed path. We then extend this idea to transform more general ARGs (containing multiple recombination cycles) into directed arborescences.

If v is a recombination node in an ARG, then by tracing two paths from the parents of v back in time, the paths eventually meet at a common ancestral, or *coalescent* node. Any pair of such directed paths together form a *recombination cycle*. Note that a recombination node v may give rise to several different recombination cycles, possibly with different coalescent nodes (Fig. 1). For a fixed recombination cycle C, denote the coalescent node of C by $coal(C)$ and the recombination node of cycle C by $rec(C)$ (Fig. 1). An *internal node* of recombination cycle C is any node in $C \backslash \{coal(C), rec(C)\}$.

For recombination cycle C, imagine placing two pebbles (labeled by a and b) on the coalescent node $coal(C)$. Consider the two node disjoint paths from the coalescent node to the recombination node (Fig. 2). We think of pebble a as traveling along one path and pebble b as traveling along the second path until they meet at the recombination node. Note that, for a fixed recombination cycle C, the only nodes simultaneously occupied by both pebbles are the coalescent node and recombination node. We enforce that, at each

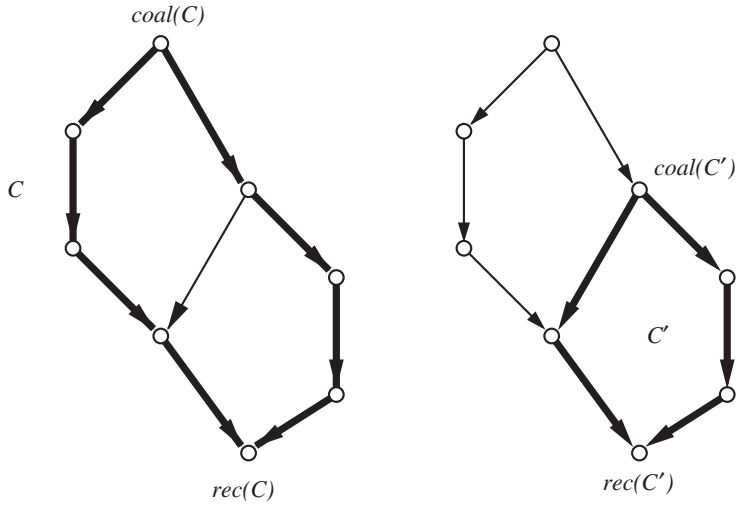

FIG. 1. Recombination cycles.

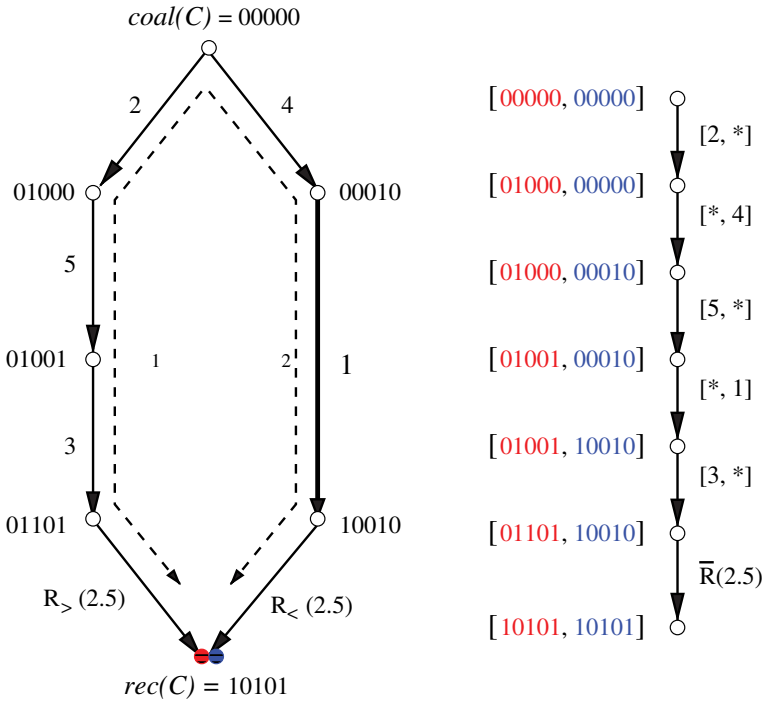

FIG. 2. Transforming a recombination cycle into a directed path.

time step, only one pebble is in motion, unless both pebbles move together to meet at the recombination node. We record the journey of the pebbles in a sequence of ordered pairs, where the first component of each ordered pair represents the position of the first pebble throughout its journey, and the second component represents the position of the second pebble throughout its journey. From the sequence of moves, it is possible to construct a graph, with ordered pairs of binary sequences as nodes and ordered pairs of mutations/recombinations labeling the edges. Note that for any recombination cycle, there may be many associated directed paths, depending on the choice of ordering for the steps taken by the pebbles. For example, pebble a could move first and then pebble b, or vice versa, resulting in two different paths. However, while there may be multiple paths associated with C, the *lengths* of all such paths are equal.

The idea outlined above can be extended to transform more general ARGs (possibly containing multiple recombination cycles) into directed arborescences. We will construct a hierarchy of representations (denoted $\mathcal{HG}_1, \mathcal{HG}_2, \mathcal{HG}_3, \ldots$) which will satisfy the following:

1. Every directed Steiner arborescence in a hierarchy graph $\mathcal{HG}_i$ can be mapped to an ARG of the same cost.
2. The set of hierarchy graphs are organized into levels, with each higher level representing a larger set of ARGs.
3. Every ARG in $\mathcal{A}$ can be transformed to a directed Steiner arborescence in $\mathcal{HG}_l$ of the same cost for a suitable level l. Furthermore, it is possible to compute an upper bound on this level l.
4. By solving a sequence of directed Steiner arborescence problems, it is possible to find a sequence of upper bounds for the minimum imperfect ARG problem that converges to the exact solution.

3.3. *Hierarchy graph: the first two levels*

We first describe in complete detail the first two levels of the hierarchy, $\mathcal{HG}_1$ and $\mathcal{HG}_2$. We assume throughout that the input is given as an $n \times m$ binary input matrix I, with rows representing sequences and columns representing varying sites.

The vertices in $\mathcal{HG}_1$ correspond to binary sequences of length m, and there is a directed edge between two vertices u_1 and u_2 if the two sequences corresponding to these vertices differ at exactly one site. $\mathcal{HG}_1$ corresponds to the hypercube in dimension m, where each undirected edge is replaced by two directed

edges, one in each direction. Hierarchy $\mathcal{HG}_1$ models homoplasy events (but not recombination events), and we have the following lemma.

Lemma 3.1. *The Directed Steiner Arborescence Problem in $\mathcal{HG}_1$ is equivalent to the Imperfect Phylogeny Reconstruction Problem.*

In level two of the hierarchy, the vertices of the graph $\mathcal{HG}_2(I)$ are ordered pairs (v_1, v_2), where v_1 and v_2 both range over binary sequences of length m. There is a directed edge from ordered pair (v_1, v_2) to (w_1, w_2) if one of the following properties is satisfied.

(I) v_1 and w_1 differ in exactly one site (site i) and $v_2 = w_2$. The edge between (v_1, v_2) and (w_1, w_2) is labeled $[i,*]$.

(II) v_2 and w_2 differ in exactly one site (site i) and $v_1 = w_1$. The edge between (v_1, v_2) and (w_1, w_2) is labeled $[*, i]$.

(III) $w_1 = w_2$ and w_1 can be obtained from v_1 and v_2 by combining the first $\lfloor k \rfloor$ sites of v_1 with the last $n - \lfloor k \rfloor$ sites of v_2. The edge between (v_1, v_2) and (w_1, w_2) is labeled $R(k)$.

(IV) $w_1 = w_2$ and w_1 can be obtained from v_1 and v_2 by combining the first $\lfloor k \rfloor$ sites of v_2 with the last $n - \lfloor k \rfloor$ sites of v_1. The edge between (v_1, v_2) and (w_1, w_2) is labeled $\overline{R}(k)$.

Edges of type (I) and (II) are called level 2 mutation edges, and edges of type (III) and (IV) are called level 2 recombination edges. Associated with the edges of the level 2 graph is a weight function $w_e : E(\mathcal{HG}(I)) \rightarrow R_{\geq 0}$, which is specified as part of the input and indicates the corresponding costs for the recombination and mutation events.

By convention, the recombination point k will always be chosen to be half-integral (e.g., $k = 2.5$ corresponds to a recombination event between sites 2 and 3), except in the following case. In order to incorporate certain types of branching events, we allow the events $R(0)$ and $\overline{R}(0)$, which indicate that the recombination node inherits the complete sequence of one of its parents. Event $R(0)$ indicates that the recombination node agrees with parent v and event $\overline{R}(0)$ indicates that the recombination node agrees with parent w. The weight of the edges corresponding to these recombination events will always be equal to zero.

Let R_2 be the node in $\mathcal{HG}_2$ corresponding to the ordered pair (r, r). For each row s of I, add the ordered pair (s, s) to the list of terminal vertices V_T. Given a solution T to the directed Steiner arborescence problem on graph $\mathcal{HG}_2$ with root R_2, weights w, and terminal vertices V_T, we now construct an imperfect ancestral recombination graph A displaying I of the same weight as T.

For any node (u, v) in T, let $F^+(u, v)$ denote the set of outgoing edges from (u, v) in arborescence T.

Initialize $Y = \{(r, r)\}$
While $Y \neq \emptyset$, let $(u, v) \in Y$
I. While $F^+(u, v) \neq \emptyset$, let $e = ((u, v), (u', v')) \in F^+(u, v)$
If the label of edge e is mutation $[i,*]$, add a directed edge from sequence u to sequence $u' = u.i$ to A
Else if the label of edge e is mutation $[*, i]$, add a directed edge from sequence v to sequence $v' = v.i$ to A
Else if the label of edge e is recombination $R(k)$, then $u' = v'$ is the sequence agreeing with u in positions less than k and agreeing with v in positions greater than k. In A, add directed edges from u to u' and from v to u'
Else if the label of the edge is recombination $\overline{R}(k)$, then $u' = v'$ is the sequence agreeing with v in positions less than k and agreeing with u in positions greater than k. In A, add directed edges from u to u' from v to u'.
Add (u', v') to Y and remove e from $F^+(u, v)$
II. **Remove** (u, v) from Y.

The result is an imperfect ARG A on input sequences I with common ancestor r (Fig. 3). The construction shows that for each Steiner arborescence, it is possible to obtain a corresponding imperfect ARG of the same cost. However, we will see in the next section that the reverse transformation is not always possible.

3.4. Crowned trees

In this section, we establish sufficient conditions on an imperfect ARG to correspond to a directed Steiner arborescence in hierarchy level 2 of the same cost. A *crowned tree* displaying I is an imperfect ARG displaying I which satisfies the following conditions.

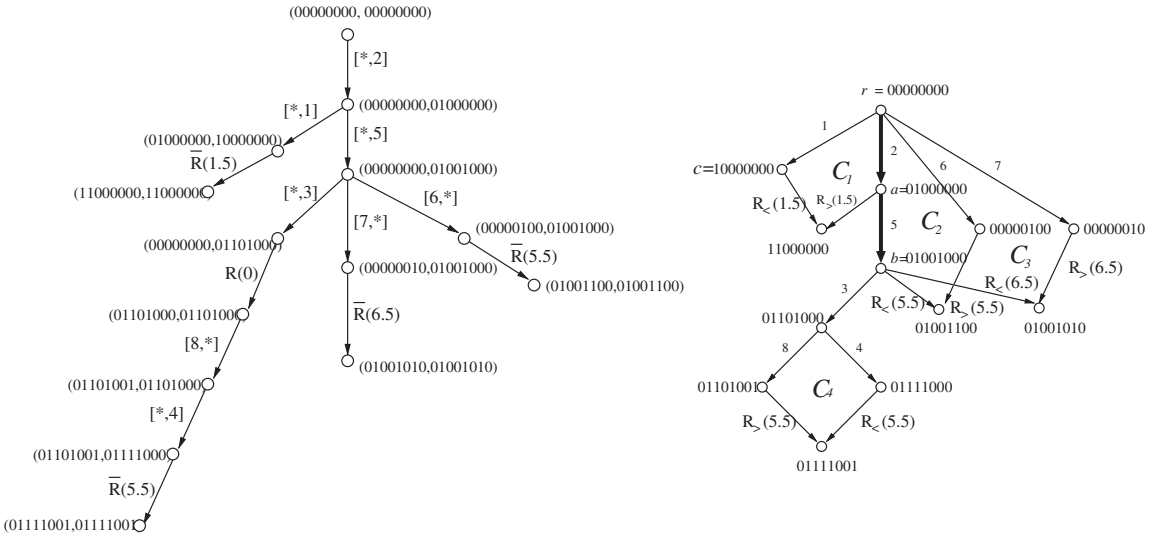

FIG. 3. Transformation from directed Steiner arborescence to imperfect ancestral recombination graph.

Condition 1. Let C and C' be two recombination cycles in the imperfect ARG. If C and C' share a node that is an internal node in at least one of C or C', then the two recombination cycles have the same coalescent node ($coal(C) = coal(C')$).

Condition 2. For any fixed recombination cycle C, let $S(C)$ denote the set of cycles C' sharing at least one node with C which appears as an internal node in C. Then there is a directed path P_C contained in C such that

i. $coal(C) \in P_C$
ii. the set of shared edges between C and any cycle $C' \in S(C)$ forms a directed path and is contained in path P_C.

Note that the recombination cycles C and C' in Figure 4 violate both conditions. A consequence of Condition 1 is that an internal node in a recombination cycle C cannot be a coalescent node for any other cycle C'. For recombination cycle C, let $last(C)$ denote the final node in path P_C, i.e., the unique node in P_C

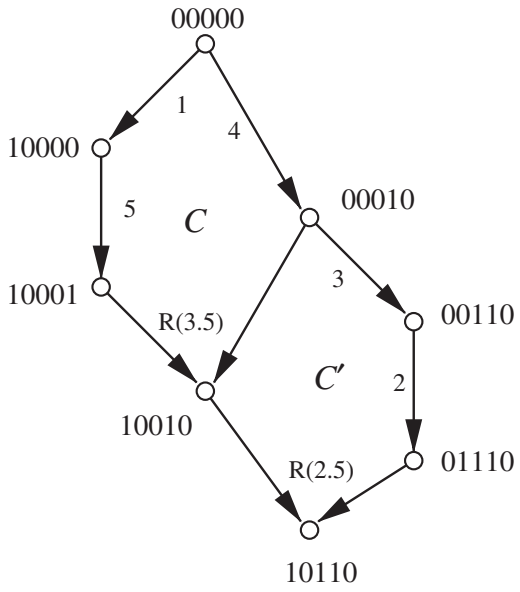

FIG. 4. An example of an imperfect ancestral recombination graph that is not a crowned tree.

with indegree 1 and outdegree 0. By condition (2), the descendants of $last(C)$ in C cannot be contained in any other recombination cycle in $S(C)$. Let $\mathcal{P}$ denote the union of directed paths P_C over all recombination cycles C. Note that $\mathcal{P}$ is a union of vertex disjoint paths, i.e., no vertex is contained in more than one path P_C. Also, observe that for each recombination cycle C, the labeling of the two directed paths ρ_1 and ρ_2 in C from $coal(C)$ to $rec(C)$ can be chosen arbitrarily. If C contains any edges in $\mathcal{P}$, then these edges must belong to exactly one of the paths ρ_1 or ρ_2 in C; by convention, we will always label this path by ρ_2.

In Figure 3, P_{C_1} is the edge (r, a) and $P_{C_2} = P_{C_3}$ is the directed path on edges (r, a) and (a, b). This establishes the paths ρ_2 for cycles C_1, C_2, and C_3 as the paths in each of these cycles containing edge (r, a); however, the paths ρ_1 and ρ_2 of cycle C_4 are not determined. Also, $last(C_1) = a$, $last(C_2) = last(C_3) = b$, and $last(C_4) = \emptyset$.

An important subset of the set of crowned trees is the set of galled trees, which are constrained ARGs in which any pair of recombination cycles are edge disjoint. This class of graphs was introduced in Wang et al. (2001) and further studied in Gusfield (2005), Gusfield et al. (2003, 2004a, b), and Song (2006). The importance of such trees was established by Gusfield et al. (2003), who develop a polynomial algorithm for the parsimonious ARG reconstruction problem over galled trees. Therefore, the set of crowned trees captures an important subfamily of ancestral recombination graphs.

We now demonstrate that any crowned tree A displaying I corresponds to a directed Steiner arborescence T of the same cost in the level 2 hierarchy graph $\mathcal{HG}_2$. The idea for building T from crowned tree A will be to visit sequences in the crowned tree in a well defined order, which will allow the pebbles to properly keep track of the positions as the sequences of the crowned tree are visited.

In our algorithm, the second pebble performs a depth first search on edges in $\mathcal{P}$. For example, in Figure 3, the second pebble first travels edge $(r, a) \in \mathcal{P}$. When the second pebble reaches a node $last(C)$ for some recombination cycle C, then the first pebble for cycle C becomes *activated* and travels down path ρ_1 of recombination cycle C. Since $a = last(C_1)$ in Figure 3, the first pebble for cycle C_1 becomes activated after the second pebble's move and travels the path containing edges (r, c) and $(c, rec(C_1))$. The intuition for the algorithm is that conditions (1) and (2) enable the first pebble to remain on the coalescent node of a cycle while the second pebble is traveling any edges contained in the common path set $\mathcal{P}$. Furthermore, the first pebble is able to visit all edges in directed path ρ_1 in C and recombine with the second pebble since no other recombination cycle intersects the vertices of ρ_1. The following algorithm describes the transformation from crowned tree to directed Steiner arborescence.

1. Consider the sequence r corresponding to the root of crowned tree A and initialize $Y = \{(r, r)\}$.
2. While $Y \neq \emptyset$
   For $y = (s, t) \in Y$
   (a) for any cycle C such that $t = last(C)$, activate s, add the set of edges in path ρ_1 to T, and add $rec(C)$ and all nodes adjacent to edges in ρ_1 to Y
   (b) if there is an outgoing mutation edge from t in A that is contained in path set $\mathcal{P}$, suppose this edge has label i. Then create a mutation edge from y to $y' = (s, t.i)$ with edge label $[*, i]$ and add y' to Y.
   (c) for each outgoing mutation edge from s (t) in A that is not contained in path set $\mathcal{P}$ with label i, create a mutation edge from y to $y' = (s.i, t)$ ($y' = (s, t.i)$) with edge label $[i, *]$ (label $[*, i]$), and add y' to Y
   (d) if s (t) has an outgoing mutation edge in A labeled i whose other endpoint is a coalescent node corresponding to sequence $s.i$ ($t.i$), create a mutation edge from y to $y' = (s.i, t)$ ($y' = (s, t.i)$), and a recombination edge from y' to $y'' = (s.i, s.i)$ labeled $\overline{R}(0)$ (from y' to $y'' = (t.i, t.i)$ labeled $R(0)$). Add y'' to Y
3. Remove $y = (s, t)$ from Y

Note that if the phylogenetic network A does not satisfy Condition (1), it would be impossible for pebble 1 to simultaneously remain on the coalescent node for all recombination cycles C. Similarly, if A does not satisfy Condition (2), it would be impossible for pebble 2 to simultaneously travel all the paths necessary before recombining with pebble 1.

3.5. Higher level hierarchy graphs

We now extend beyond the first two levels to construct a hierarchy of representations, with each successive level in the hierarchy representing larger and larger subsets of ARG. The kth level of the hierarchy

will consist of an underlying graph $\mathcal{HG}_k$, whose vertices are k-dimensional vectors $(v_1, v_2, \ldots v_k) \in (\mathbb{Z}_2^m)^k$ where each coordinate v_i ranges over all binary sequences of length m. In the context of our informal presentation based on pebble motions, each coordinate corresponds to a different pebble and pairs of coordinates correspond to pairs of pebbles a and b that are allowed to traverse paths which form recombination cycles. The directed edges of $\mathcal{HG}_k$ will detail the possible transition steps the pebbles are allowed to make as they travel through the imperfect ARG. For a vector $v \in \mathbb{R}^k$ and a subset $C \subseteq [k]$ of coordinates, let $v|_C$ denote the restriction of vector v onto coordinates in C. We will consider $v|_C$ as a set (and not a multi-set), so that the set $v|_C$ has size one if all of the coordinates of v in positions C are the same. Let l be either zero or a half integral value between $\frac{1}{2}$ and $m - \frac{1}{2}$. For two binary sequences a and b of length m, let $(a, b)_l$ denote the sequence obtained by combining the first $\lfloor l \rfloor$ sites of a with the final $m - \lfloor l \rfloor$ sites of b (in the case $l = 0$, $(a, b)_0 = b$). To describe the directed edges in hierarchy graph $\mathcal{HG}_k$, we will need to define the following operations.

(i) For a vector v, suppose there is a subset of coordinates $C \subseteq [k]$ containing the same binary sequence a. Then the *coordinated mutation transition* at site i on C results in vector $M_{C,i}(v) \in (\mathbb{Z}_2^m)^k$, whose jth coordinate is

$$M_{C,i}(v)_j = \begin{cases} a.i \text{ if } j \in C \\ v_j \text{ if } j \notin C \end{cases}$$

This move corresponds to modifying vector v by taking the sequences with coordinates in C and mutating site i in each of these sequences. Note that $M_{C,i}(v)|_C = \{a.i\}$, and furthermore, there may be additional coordinates in $[k] \backslash C$ for which v contains sequence a and remains unchanged by this transition.

(ii) Consider any two disjoint sets of coordinates $C_1, C_2 \subset [k]$ such that $v|_{C_1} = \{a\}$ and $v|_{C_2} = \{b\}$ (where $a \neq b$) Consider two subsets $C_1' \subseteq C_1$ and $C_2' \subseteq C_2$, one (but not both) of which may be the empty set and let l be either 0 or a half integral value between $\frac{1}{2}$ and $m - \frac{1}{2}$. Then the jth coordinate of *recombination transitions* $R_{C_1, C_2, C_1', C_2', l}(v)$ and $\overline{R}_{C_1, C_2, C_1', C_2', l}(v)$ are defined as

$$R_{C_1, C_2, C_1', C_2', l}(v)_j = \begin{cases} (a, b)_l \text{ if } j \in C_1' \cup C_2' \\ v_j \text{ otherwise} \end{cases}$$

$$\overline{R}_{C_1, C_2, C_1', C_2', l}(v)_j = \begin{cases} (b, a)_l \text{ if } j \in C_1' \cup C_2' \\ v_j \text{ otherwise.} \end{cases}$$

The transitions (i) and (ii) allow us to define the edges of $\mathcal{HG}_k$.

(E1) For a vector of binary sequences $v \in (\mathbb{Z}_2^m)^k$, for each integer $i \in [m]$, and for each subset $C \subseteq [k]$ such that $|v|_C| = 1$, there is a directed edge from v to $M_{C,i}(v)$.

(E2) For a vector of binary sequences $v \in (\mathbb{Z}_2^m)^k$, $l \in \{0, \frac{1}{2}, \frac{3}{2}, \ldots m - \frac{1}{2}\}$, and for each $C_1, C_2 \subseteq [k]$ and subsets $C_1' \subseteq C_1, C_2' \subseteq C_2$ such that $C_1 \cap C_2 = \emptyset$ and $|v|_{C_1}| = |v|_{C_2}| = 1$, there is a directed edge from v to $R_{C_1, C_2, C_1', C_2', l}(v)$ and from v to $\overline{R}_{C_1, C_2, C_1', C_2', l}(v)$.

Edges of type (E1) are called hierarchy graph mutation edges, and edges of type (E2) are called hierarchy graph recombination edges. Note that different choices of subsets C_1 and C_2 and values l in the recombination transitions may in fact give rise to the same hierarchy graph recombination edge. Together with the edges of the hierarchy graph is an associated weight function $w_e : E(\mathcal{HG}_k) \to \mathbb{R}_{\geq 0}$, which is specified as part of the input and indicates the corresponding costs for the recombination and mutation events. In general, the weight function can be site-dependent or chosen according to existing information about recombination and mutation frequencies in the population. In the *uniform* model, all recombination events (except recombination events corresponding to $l = 0$) have the same cost α_r and all mutation events have the same cost α_m. This corresponds to assigning weight α_r to all hierarchy graph recombination edges, weight α_m to all hierarchy graph mutation edges, and weight zero to the remaining edges.

Now, suppose D is a solution to the minimum Steiner arborescence problem on graph $\mathcal{HG}_k$ with root $Root_k$, weights w, and terminal vertices V_T. We will describe a map Φ_k which constructs from D an imperfect ancestral recombination graph $A = \Phi_k(D)$. For any node u in D, let $F^+(u)$ denote the set of outgoing edges from u in arborescence D. The transformation describes a breadth first search through the set of vertices in D, with each explored edge giving rise to a set of edges in the imperfect ARG A. In the following description, Y will denote the vertices in D which have outgoing edges remaining to be explored. For each k, the map Φ_k transforms each Steiner arborescence in $\mathcal{HG}_k$ to an imperfect ARG.

Definition 1. *An imperfect ARG A is* representable at level *k of the ARG hierarchy if there exists a Steiner arborescence D in* $\mathcal{HG}_k$ *such that* $\Phi_k(D) = A$. *The* ARG-width *of A is the smallest k such that A is representable at hierarchy level k.*

INPUT: DIRECTED STEINER ARBORESCENCE $D \in \mathcal{HG}_k$
OUTPUT: IMPERFECT ANCESTRAL GRAPH $A = \Phi_k(D)$ WITH VERTEX SET $V(A)$ AND EDGE SET $E(A)$
Initialize $V(A) = \{r\}, Y = \{(r, r, \ldots r)\}$
While $Y \neq \emptyset$, let $u \in Y$
I. While $F^+(u) \neq \emptyset$, let $e = (u, v) \in F^+(u)$
If the label of edge e is hierarchy mutation edge corresponding to coordinates C and site i, add $M_{C,i}(u)$ to $V(A)$, and add directed edge from sequence u to sequence $u' = u.i$ to $E(A)$
Else if edge e is hierarchy recombination edge labeled by $R_{C_1,C_2,C_1',C_2',l}(v)$, where $u|_{C_1} = \{a\}$ and $u|_{C_2} = \{b\}$, then add sequence $(a, b)_l$ to $V(A)$, and add directed edges from a to $(a, b)_l$ and from b to $(a, b)_l$ to $E(A)$
Else if edge e is hierarchy recombination edge labeled by $\overline{R}_{C_1,C_2,C_1',C_2',l}(v)$, where $u|_{C_1} = \{a\}$ and $u|_{C_2} = \{b\}$, then add $(b, a)_l$ to $V(A)$, and add directed edges from a to $(b, a)_l$ and from b to $(b, a)_l$ to $E(A)$
Add v to Y and **remove** e from $F^+(u)$
II. **Remove** u from Y.

By Section 3.4, we have the following lemma.

Lemma 3.2. *The set of crowned trees has ARG-width equal to two.*

We now study the structure of representable imperfect ancestral recombination graphs.

Lemma 3.3. *If an imperfect ARG A is representable at level k, then it is representable at level k′ for all* $k' \geq k$.

Proof. Since A is representable at level k, there exists an arborescence $D_k \in \mathcal{HG}_k$ such that $\Phi_k(D_k) = A$. Let $Root_k = (\underbrace{r, r, \ldots r}_{k}) \in \mathcal{HG}_k$ and let $Root_{k'} = (\underbrace{r, r, \ldots r}_{k'}) \in \mathcal{HG}_{k'}$. Let $D_{k'}$ be the directed Steiner arborescence in $\mathcal{HG}_{k'}$ obtained by appending $k' - k$ copies of binary sequence r to each vertex in D_k. Now, if s is either the root of D or an input row in I, the vector

$$v(s) = (\underbrace{s, s, \ldots s}_{k}, \underbrace{r, \ldots, r}_{k'-k})$$

appears in $D_{k'}$. Furthermore, for each input sequence s, the path in D_k between $Root_k$ and s gives rise to a path in $D_{k'}$ between $Root_{k'}$ and $v(s)$ (with each vertex in the path having value r in the final $k' - k$ entires). Now, for each input sequence s, create a trivial recombination edge between $v(s)$ and the vector

$$R_{([k],[k']-[k],[k],[k']-[k],0)}(v(s)) = (\underbrace{s, s, \ldots s}_{k'}) \in \mathcal{HG}_{k'}.$$

The resulting graph is a Steiner arborescence D' in $\mathcal{HG}_{k'}$. Since trivial recombination edges have weight zero, the directed Steiner arborescence D' has the same cost as the directed Steiner arborescence D. Furthermore $\Phi_{k'}(D_{k'}) = A$, implying that A is representable at level k'. ■

For fixed values of the mutation and recombination parameters, let $\{D_k^*\}_{k\geq 1}$ denote a sequence of solutions to the MDSA problem on the sequence of hierarchy graphs $\{\mathcal{HG}_k\}_{k\geq 1}$. We apply the transformations Φ_k to these arborescences to obtain a sequence of imperfect ARGs $\{\Phi_k(D_k^*)\}_{k\geq 1}$. The following is a corollary of Lemma 3.3.

Corollary 3.4. *For any* $k \geq 1$, $cost(D_k^*) \geq cost(D_{k+1}^*)$.

For an imperfect ARG A, let $R(A)$ denote the number of recombination events in A. Note that $R(A)$ simply counts the number of recombination events (not the weighted cost of recombinations) and does not take into account any homoplasy events in A. The following theorem bounds the ARG-width of any ARG.

Theorem 3.5. *For any imperfect ancestral recombination graph A with* $R(A) \geq 1$, *the ARG-width of A is at most* $2R(A)$.

Proof. Let A be an imperfect ARG with at least one recombination node. Our goal is to construct a Steiner arborescence S in $\mathcal{HG}_{2R(A)}$ such that $\Phi_{2R(A)}(S) = A$.

We begin by using A to construct a set of directed paths. For each recombination node y in A, we will consider two directed paths $p_1(y)$ and $p_2(y)$ from the root r to node y. Furthermore, for any non-recombination node y in A, we will define a single directed path $p_1(y)$ from the root to node y. These paths are defined inductively as follows. For root r, the path $p_1(r)$ is the trivial path containing the single vertex r. Suppose y is a node all of whose parental paths have been constructed. If node y is a recombination node with parents u and v, then path $p_1(y)$ is obtained by taking the path $p_1(u)$ together with edge (u, y) and path $p_2(y)$ is obtained by taking the path $p_1(v)$ together with edge (v, y). Note that different choices for labeling the parent nodes u and v possibly lead to different sets of paths; in such a case, we can make these choices arbitrarily. If node y is not a recombination node, then it has a single parent u and path $p_1(y)$ is obtained by taking the path $p_1(u)$ together with edge (u, y).

Now, let $y_1, y_2, \ldots y_{R(A)}$ denote the set of recombination nodes in A and let $\mathcal{P}$ be the set of $2R(A)$ paths $p_1(y_i)$ and $p_2(y_i)$ for $1 \leq i \leq R(A)$. These paths will map the journey of $2R(A)$ pebbles through the graph, corresponding to the $2R(A)$ coordinates of vertices in $\mathcal{HG}_{2R(A)}$. The journey of these pebbles will determine the Steiner arborescence D in $\mathcal{HG}_{2R(A)}$. For each edge $e = (u, v)$, let $\mathcal{P}_e$ denote the set of paths in $\mathcal{P}$ which contain edge e and let C_e denote the coordinates of the pebbles corresponding to these paths.

We enforce that for each edge, all the pebbles whose paths $\mathcal{P}_e$ intersect e must traverse this edge *simultaneously* in the Steiner arborescence D. The construction will build the Steiner arborescence from the leaves up to the root. In each stage of the transformation, an *activated node* v in A is a node such that all descendants of v have already been considered. In the following transformation, Q denotes the set of activated nodes, Z denotes the set of nodes in A waiting to be activated, and α is a map that takes each node in A to a node in D.

INPUT: IMPERFECT ANCESTRAL RECOMBINATION GRAPH A

OUTPUT: DIRECTED STEINER ARBORESCENCE $D \in \mathcal{HG}_{2R(A)}$ SUCH THAT $\Phi_{2R(A)}(D) = A$

1. Let r be the root of A and initialize
   $Q = \{y : y \text{ is a node in } A \text{ with no descendants}\}$, $Z = Q, \alpha(y) = (y, y, \ldots y)$ for each $y \in Q, V(D) = \{\alpha(y) : y \in Q\}$, $E(D) = \emptyset$.
2. While $Q \neq \emptyset$
   For $y \in Q$
   (a) If $E^-(y)$ has an edge (x, y) labelled by a mutation event i, remove y from Q and add x to Z. Let $\alpha(x) = M_{C_{(x,y)},i}(\alpha(y))$, add vertex $\alpha(x)$ to $V(D)$, and add the edge $(\alpha(x), \alpha(y))$ to $E(D)$
   (b) If y is a recombination node in A, then there are two recombination edges (x_1, y) and (x_2, y) in $E^-(y)$. Let C_1 denote the set of paths in P using edge (x_1, y), let C_2 denote the set of paths in $\mathcal{P}$ using edge (x_2, y) (note that these paths are disjoint), and let $l \in \{0, \frac{1}{2}, \ldots m - \frac{1}{2}\}$ such that $y = (x_1, x_2)_l$.
   Let

$$\alpha(x_1)_i = \alpha(x_2)_i = \begin{cases} \alpha(y)_i \text{ for } i \notin C_1 \cup C_2 \\ x_1 \text{ for } i \in C_1 \\ x_2 \text{ for } i \in C_2 \end{cases}$$

   Remove y from Q, add x_1 and x_2 to Z, add vertices $\alpha(x_1)$ and $\alpha(x_2)$ to $V(D)$ and add edges $(\alpha(x_1), \alpha(y))$ and $(\alpha(x_2), \alpha(y))$ to $E(D)$.
   (c) if $x \in Z$ has no descendants in Z, then remove x from Z and add x to Q.

This resulting directed Steiner Arborescence together with the map α shows imperfect ARG A has ARG-width bounded above by $2R(A)$. ■

For a fixed set of mutation and recombination parameters, let $\mathcal{M}(I)$ denote the set of minimum imperfect ARGs for input set I and let

$$R^*_{min}(I) = \min_{A \in \mathcal{M}(I)} R(A).$$

We first extend the notion of representability to sets of input sequences.

Definition 2. *An input set of binary sequences I is* representable *at level k of the ARG hierarchy if there exists a Steiner arborescence S in $\mathcal{HG}_k$ and an imperfect ARG A displaying I such that $\Phi_k(S) = A$. The* ARG-width *of I is the smallest k such that I is representable at level k.*

The following is now a corollary of Theorem 3.5.

Corollary 3.6. *For any input set I, the ARG-width of I is at most $2R^*_{min}(I)$. It follows that the minimum imperfect ARG reconstruction problem can be solved to optimality at hierarchy level $2R^*_{min}(I)$.*

4. EXPERIMENTAL RESULTS

In practice, since we do not know the value of $R^*_{min}(I)$, we cannot determine how high in the hierarchy we would need to go in order to solve the minimum imperfect ARG reconstruction problem to optimality. In this section, we concentrate on level 2 of the hierarchy and design algorithms and heuristics to solve the MDSA problem in $\mathcal{HG}_2$. In practice, Steiner arborescences in $\mathcal{HG}_2$ can be efficiently constructed for benchmark and simulated data sets. It is then possible to compare the upper bounds on the number of events with lower bounds (and if possible, with optimal solutions) for the ARG reconstruction problem, which considers recombination events only in the infinite sites model.

We analyze the performance on a benchmark data set as well as on simulated data sets. The implementation was performed in C++ and solved using CPLEX 11; tests were conducted on a Athlon 64 Dual Core 2GHz Processor with 2G RAM, running Linux.

The software implementing our algorithm, iARG, is available for download at `www.cs.brown.edu/people/sorin/lab/pages/software.html`.

4.1. Linear programming formulation

A common approach for studying the MDSA problem is to use integer and linear programming methods. We implemented a well-known linear programming formulation for the MDSA problem, detailed in Sridhar et al. (2007b).

We applied the linear program to several sets of simulated data obtained by the MS program of Hudson Hudson (2002) and the `seq-gen` program of Rambaut and Grassly (1997). Using these exact methods, we were able to solve instances of up to 11 sites, with varying number of individuals.

4.2. Steiner tree heuristics

Since we would like to develop methods that scale to solve larger instances, we apply known heuristics for the directed Steiner arborescence problem. We implement the following insertion heuristic.

Initialize $T = \{(r, r)\}$ (where r denotes the common ancestor sequence) and $N_T = V_T$
While $N_T \neq \emptyset$
  Find nodes $u^* \in T$ and $v^* \in N_T$ minimizing the distance from T to the set N_T
  Add the nodes and arcs of the shortest path from u^* to v^* to tree T
  Remove v^* from N_T

This algorithm is known to have worst-case error ratio equal to the number of input terminals V_T (Voss, 1993). We applied this heuristic to the well-studied benchmark *Drosophila melanogaster* data set from Kreitman (1983). This data set was previously studied in the context of ARG reconstruction in Bafna and Bansal (2004) and Song and Hein (2003, 2005).

The size of the original data is 11 haplotypes, each of length 2800 base pairs. We perform the same operations as Bafna and Bansal (2004), deleting identical haplotypes, removing all sites that are pairwise compatible with all other sites, and removing sites that are identical to adjacent sites. This results in an input matrix of nine haplotypes of length 16. Our heuristic solved the resulting reduced problem in 10.8 seconds, giving four recombination events and three back mutations under the uniform model (Fig. 5). Furthermore, the output of the algorithm shows the four recombinations occuring in only two positions, with two recombination events between sites 4 and 5, and two recombination events between sites 5 and 6.

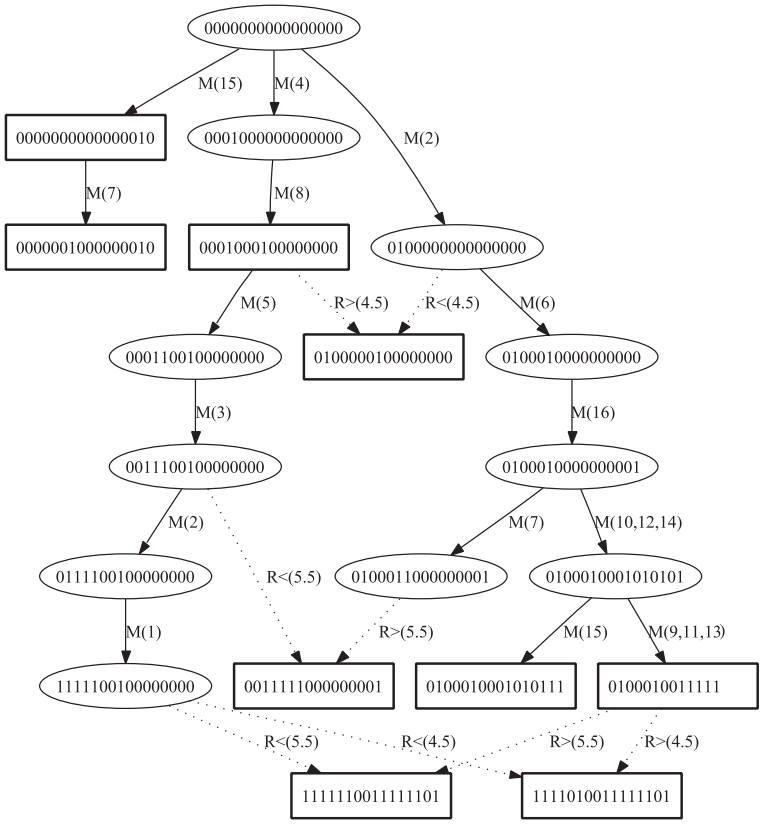

FIG. 5. Output of minimum directed steiner arborescence heuristic on Kreitman Drosophila data.

In Song and Hein (2005), it is shown that the minimum number of recombinations for this data set under the infinite sites model is $R_{min}(I) = 7$. Therefore, our algorithm quickly finds a solution with as many recombinations plus back/recurrent mutations as the number of recombinations only in the optimal solution for the parsimonious ARG reconstruction problem.

5. CONCLUSION

We have introduced and developed a framework for solving the imperfect ARG reconstruction problem. This unifies the current models for ARG reconstruction and imperfect phylogeny reconstruction into a single framework. There are many potential avenues for future work, including the following:

1. Extending the model to handle missing data
2. Finding ways to restrict the vertices in the hierarchy graphs that need to be searched for the optimal Steiner tree, in the spirit of Buneman graphs (Semple and Steel, 2003)
3. Extending the model to include other types of recombination and hybridization events

We plan to explore these avenues and improve the algorithms presented to generate provably optimal ARGs from variation data. Such analyses are increasingly important, as data sets for larger populations are gathered.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Agarwala, R., and Fernandez-Baca, D. 1994. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM J. Comput.* 23, 1216–1224.

Bafna, V., and Bansal, V. 2004. The number of recombination events in a sample history: conflict graph and lower bounds. *IEEE Trans. Comput. Biol. Bioinform.*, 1, 78–90.

Bandelt, H. 1991. Phylogenetic networks. *Verh. Naturwiss. Ver. Hamburg* 34, 5157.

Blelloch, G.E., Dhamdhere, K., Halperin, E., et al. 2006. Fixed parameter tractability of binary near-perfect phylogenetic tree reconstruction. *Lect. Notes Comput. Sci.* 4051, 667–678.

Bordewich, M., and Semple, C. 2007. Computing the minimum number of hybridisation events for a consistent evolutionary history. *Discrete Appl. Math.* 155, 914–928.

Damaschke, P. 2004. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Lect. Notes Comput. Sci*. 3162, 1–12.

Foulds, L.R., and Graham, R.L. 1982. The Steiner problem in phylogeny is NP-complete. *Adv. Appl. Math.* 3, 43–49.

Ganapathy, G., Ramachandran, V., and Warnow, T. 2003. Better hill-climbing searches for parsimony. *Lect. Notes Comput. Sci.* 2812, 245–258.

Griffiths, R.C., and Marjoram, P. 1997. An ancestral recombination graph, 257–270. *In* Donnelly, P., and Tavaré, S., eds. *Progress in Population Genetics and Human Evolution. IMA Volumes in Mathematics and its Applications. Volume 87.* Springer Verlag, Berlin.

Guha, S., and Khuller, S. 1996. Approximation algorithms for connected dominating sets. *Eur. Symp. Algorithms* 179–193.

Gusfield, D. 1991. Efficient algorithms for inferring evolutionary trees. *Networks* 21, 19–28.

Gusfield, D. 1999. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Gusfield, D. 2005. On the full-decomposition optimality conjecture for phylogenetic networks. [Technical report]. University of California, Davis.

Gusfield, D., Eddhu, S., and Langley, C. 2003. Efficient recombination of phylogenetic networks with constrained recombination. *Proc. IEEE Comput. Soc. Bioinform. Conf.* 363–374.

Gusfield, D., Eddhu, S., and Langley, C. 2004a. The fine structure of galls in phylogenetic networks. *INFORMS J. Comput.*16, 459–469.

Gusfield, D., Eddhu, S., and Langley, C. 2004b. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinform. Comput. Biol.* 2, 173–213.

Hein, J. 1990. Reconstructing evolution of sequences subject to recombination using parsimony. *Math. Biosci.* 98, 185–2000.

Hein, J. 1993. A heuristic method to reconstruct the history of sequences subject to recombination. *J. Mol. Evol.* 36, 296–405.

Hudson, R. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18, 337–338.

Hudson, R., and Kaplan, N. 1985. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics* 11, 147–164.

Hwang, F., Richards, D., and Winter, P. 1992. *The Steiner Tree Problem in Graphs*. North-Holland, Amsterdam.

International HapMap Consortium. 2005. The International HapMap project. `www.hapmap.org.` *Nature* 426, 789–796.

Karp, R. 1972. Reducibility among combinatorial problems, 85–103. In: Miller, R.E., Thatcher, J.W., eds. *Complexity of Computer Computations*. Plenum Press, New York.

Kreitman, M. 1983. Nucleotide polymorphism at the alcohol dehydrogenase locus of *Drosophila melanogaster. Nature* 304, 412–417.

Lyngs, R., Song, Y., and Hein, J. 2005. Minimum recombination histories by branch and bound. *Lect. Notes Comput. Sci.* 3692, 239–250.

Meyers, S., and Griffiths, R. 2003. Bounds on the minimum number of recombination events in a sample history. *Genetics* 163, 375–394.

Nordborg, M. 2001. *Coalescent Theory*. John Wiley & Sons, New York.

Rambaut, A., and Grassly, N. 1997. Seq-gen: an application for the Monte-Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.* 13, 235–238.

Semple, C., and Steel, M. 2003. *Phylogenetics. Mathematics and Its Applications Series, No. 22*. Oxford University Press, New York.

Song, Y. 2006. A concise necessary and sufficient condition for the existence of a galled-tree. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 3, 186–191.

Song, Y., and Hein, J. 2003. Parsimonious reconstruction of sequence evolution and haplotype blocks. *Lect. Notes Comput. Sci.* 2812, 287–302.

Song, Y., and Hein, J. 2004. On the minimum number of recombination events in the evolutionary history of DNA sequences. *J. Math. Biol.* 48, 160–186.

Song, Y., and Hein, J. 2005. Constructing minimal ancestral recombination graphs. *J. Comput. Biol.* 12, 147–169.

Song, Y.S., Wu, Y., and Gusfield, D. 2005. Algorithms for imperfect phylogeny haplotyping (IPPH) with a single homoplasy or recombination event. *Lect. Notes Comput. Sci.* 3692, 152–164.

Sridhar, S., Dhamdhere, K., Blelloch, G.E., et al. 2006. Simple reconstruction of binary near-perfect phylogenetic trees. *Lect. Notes Comput. Sci.* 3992, 799–806.

Sridhar, S., Dhamdhere, K., Blelloch, G.E., et al. 2007a. Algorithms for efficient near-perfect phylogenetic tree reconstruction in theory and practice. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 4, 561–571.

Sridhar, S., Lam, F., Blelloch, G. E., et al. 2007b. Efficiently finding the most parsimonious phylogenetic tree via linear programming. *Lect. Notes Comput. Sci.* 4463, 37–48.

Voss, S. 1993. Worst-case performance of some heuristics for Steiner problem in directed graphs. *Inform. Process. Lett.* 48, 99–105.

Wang, L., Zhang, K., and Zhang, L. 2001. Perfect phylogenetic networks with recombination. *J. Comput. Biol.* 8, 67–78.

Winter, P. 1987. Steiner problem in networks, a survey. *Networks* 17, 129–167.

Address correspondence to:
Dr. Fumei Lam
Department of Computer Science
Brown Univerisity
115 Waterman Street, Box 1910
Providence, RI 02912

E-mail: lam@cs.brown.edu