JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 11, 2010

Pp. 1561–1573
DOI: 10.1089/cmb.2009.0268

Constructing Consensus Genetic Maps in Comparative Analysis

XIN CHEN and JIAN-YI YANG

ABSTRACT

The construction of consensus genetic maps is a very challenging problem in computational biology. Many computational approaches have been proposed on the basis of only the marker order relations provided by a given set of individual genetic maps. In this article, we propose a comparative approach to constructing consensus genetic maps for a genome, which further takes into account the order relations from a closely related genome when resolving ordering conflicts among individual genetic maps. It aims to retain as many order relations as possible from individual genetic maps while achieving the minimum rearrangement distance to the reference genome. We implement the proposed approach as an integer linear program and test it on both simulated and real biological datasets. The experimental results show that it is capable of constructing more accurate consensus genetic maps than the most recent approach called MERGEMAP.

Key words: breakpoint distance, comparative genomics, consensus genetic map, integer linear programming.

1. INTRODUCTION

IN RECENT YEARS, the rapid adoption of high-throughput genotyping technologies such as recombination analysis and physical imaging makes multiple genetic maps available for a same species. Combining these maps into a consensus genetic map allows us to produce a higher density of markers and therefore a greater genome coverage than any individual genetic map. However, there often exist order conflicts among these individual genetic maps, mostly due to experimental errors. Constructing consensus genetic maps is thus a very challenging task in computational biology.

Many computational approaches have been proposed in the literature to construct consensus genetic maps. For example, the software JOINMAP implemented a statistical approach based on data pooling (Stam, 1993; Jansen et al., 2001). It first estimates pairwise marker distances by weighted least squares and then performs a numerical search for the best fitting orders of markers. Yap et al. (2003) proposed a graph-theoretic model, which represents individual genetic maps as directed acyclic graphs (DAGs) and merges them into a single directed graph. A directed cycle in the resulting graph hence indicates an ordering conflict among the individual genetic maps. In order to resolve ordering conflicts, Jackson et al. (2005, 2008) proposed to break the cycles by removing a minimum weighted set of feedback edges, while

Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

Wu et al. (2008a,b) adopt a better strategy which aims to remove a minimum weighted set of *feedback marker occurrences* from the individual genetic maps to be combined.

Observe that all the preceding approaches rely solely on the mapping information provided by a given set of individual genetic maps for a species. In this article, we propose a comparative approach to constructing consensus genetic maps, which further takes into account the order relations from a closely-related species whose markers are already in total order. Through a comparative analysis we might be able to identify erroneous orders among the set of conflicting orders in a biologically meaningful way, which is often seen impossible with the individual genetic maps alone. Hence, our proposed approach is particularly aimed at finding a consensus genetic map for a species, which retains as many order relations as possible from an input set of individual genetic maps, while achieving the lowest possible rearrangement distance to a closely-related species whose marker orders are already fully identified.

There have been quite a few comparative studies conducted for genetic maps. For example, Zheng et al. (2005) presented the first study on the problem of computing the rearrangement distance between consensus genetic maps of two related species. Blin et al. (2007) studied a restricted version of this problem where one of the maps is given in a total order of markers. More algorithmic studies on both problems have also been conducted (Chen and Cui, 2009; Fu and Jiang, 2007). In another study by Zheng et al. (2007), a comparative approach was proposed to reconstruct synteny blocks between two genetic maps by eliminating as few noisy markers as possible. The most recent study, and also most related to our present study, refines genetic maps by accounting for the phylogenetic information of a species tree (Bertrand et al., 2008, 2009). It aims to resolve as many incompatible marker pairs as possible, but may not end up with a conflict-free genetic map. Note that unlike our present study none of them are devoted to constructing consensus genetic maps.

The rest of the article is organized as follows. We first introduce some preliminary facts and definitions in Section 2. Our proposed approach to constructing consensus genetic maps is presented in Section 3, and its ILP formulations are developed in Section 4. Section 5 presents the experimental results on both simulated and real biological data. Finally, some concluding remarks are made in Section 6. For the sake of consistency, we borrowed many notations from Fu and Jiang (2007) and Wu et al. (2008b) throughout the article.

2. PRELIMINARIES

2.1. Individual genetic maps

An *individual genetic map* is a linear sequence of *bins*, each of which may contain one or several genetic markers. It is generated from a single mapping study and defines a partial order on markers of a chromosome. Markers from different bins are ordered by their respective bins, but for markers in the same bin their relative orders are undetermined. Consider, for example, the genetic map 2 {5 6} 4 3, where a curly bracket encloses two markers from a same bin. Both markers 5 and 6 are ordered immediately after marker 2, but no relative order is provided between markers 5 and 6. A genetic map is often modelled as a so-called *map graph*, that is, a directed acyclic graph (DAG) whose vertices represent markers and edges connect markers only from adjacent bins (Yap et al., 2003). See Figure 1 for the map graph of the above-mentioned genetic map. Note that two markers are ordered in an individual genetic map if and only if there is a directed path between them in the corresponding map graph. We denote an individual genetic map by Π, which, without ambiguity, refers to both a linear sequence of bins and its DAG representation (i.e., the map graph). The set of markers in Π is further denoted by V_Π, and the set of edges by E_Π.

When a chromosome has multiple genetic maps available from different mapping studies, one might be able to determine relative orders for more marker pairs through map integration. To this end, a directed weighted graph (DWG), termed *aggregate graph*, is often used to present all the possible (not necessarily consistent) marker orders. It is commonly constructed by taking the set union of markers and edges from all the individual genetic maps and weighting each edge by the number of times it appears in the individual genetic maps to be combined. As in map graphs, two markers are deemed to be ordered if and only if there is a directed path connecting them in the aggregate graph. However, there might exist order conflicts between individual maps, each giving rise to a directed cycle in the aggregate graph. For example, the chromosome Ω in Figure 1 has two individual genetic maps Π_1 and Π_2, which we denote by $\Omega = \{\Pi_1, \Pi_2\}$. Its DWG representation (i.e., the aggregate graph), also denoted by Ω, contains a cycle between markers

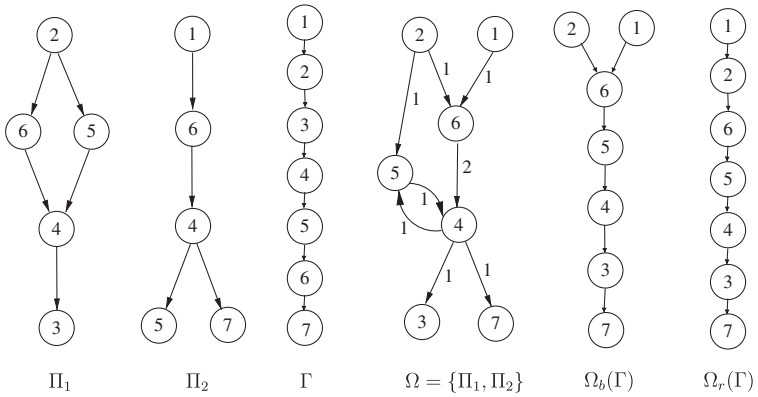

FIG. 1. An illustrative example of constructing consensus genetic maps in comparative analysis.

4 and 5, indicating an order conflict between these two markers—marker 4 precedes marker 5 in the map Π_1 but succeeds it in the other map Π_2.

2.2. *Consensus genetic maps*

The primary purpose of integrating multiple genetic maps is to resolve order conflicts between individual maps and derive additional inferences about marker orders. The end result of a map integration exercise is typically called a *consensus genetic map*. It defines a partial order on markers with higher coverage and accuracy than any component individual genetic map (Jackson et al., 2005; Yap et al., 2003). Like an individual genetic map, a consensus genetic map can also be modelled as a directed acyclic graph, but often requiring a general graphical structure. In comparison, the DAG representation of an individual genetic map has a much simpler graphical structure because unordered markers are limited to the same bin. Many approaches for map integration have been proposed in the literature, and vary mainly in the objective function to be optimized for computing a consensus genetic map. A commonly used approach is to find a consensus genetic map that is an acyclic subgraph induced from the aggregate graph by removing the smallest set of edges called the *minimum feedback edge set* (Fig. 2).

In practice, geneticists are used to working with genetic maps that are in a linear order of marker bins, and often find consensus genetic maps too complex to be convenient for many subsequent genetic analyses such as rearrangement analysis (Wu et al., 2008b; Yap et al., 2003). Therefore, it is desirable to impose some restrictions on the graphical structure of consensus genetic maps. In this study, a consensus genetic map refers to particularly a linear sequence of bins of markers in which an unordered pair of markers occurs only in the same bin, just as seen in an individual genetic map. Note that this might be the graphical structure that can closest represent the true map, if not a total order.

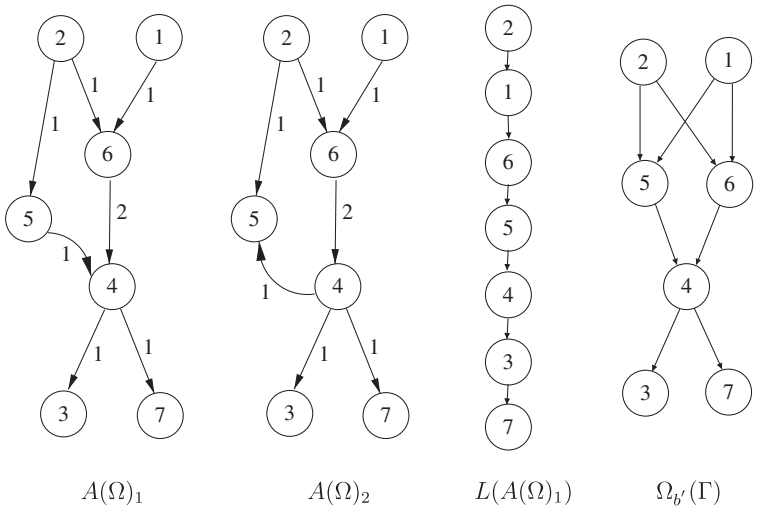

FIG. 2. $A(\Omega)_1$ and $A(\Omega)_2$ are two maximum weighted acyclic subgraphs of Ω, which is given in Figure 1. $L(A(\Omega)_1)$ and $\Omega_{b'}(\Gamma)$ are respective consensus genetic maps (without removing spurious orders) returned by the two ILP algorithms with the input aggregate graph Ω.

2.3. Rearrangement distances between partially ordered genomes

In comparative genomics studies, the *breakpoint distance* and the *reversal distance* are commonly used as the evolutionary distance between two genomes. They were initially defined on two genomes in the total order, and recently generalized to two partially ordered genomes by introducing the concept of linearization (Fu and Jiang, 2007). A *linearization* of a partially-ordered genome Π is a topological sort that represents a possible total order of all markers in Π. Let $\mathcal{L}(\Pi)$ be the set of all possible linearizations of Π.

Given two partially ordered genomes Π and Γ that have the same content of markers, their breakpoint distance and reversal distance, denoted, respectively, by $d_b(\Pi, \Gamma)$ and $d_r(\Pi, \Gamma)$, are then defined (Fu and Jiang, 2007) as

$$d_b(\Pi, \Gamma) = \min_{\pi \in \mathcal{L}(\Pi),\ \gamma \in \mathcal{L}(\Gamma)} d_b(\pi, \gamma),$$

and

$$d_r(\Pi, \Gamma) = \min_{\pi \in \mathcal{L}(\Pi),\ \gamma \in \mathcal{L}(\Gamma)} d_r(\pi, \gamma),$$

where $d_b(\pi, \gamma)$ and $d_r(\pi, \gamma)$ denote the breakpoint distance and the reversal distance between two permutations π and γ, respectively. For example, if $\Pi = 1\ 2\ 3\ 4\ 5\ \{6\ 7\}$ and $\Gamma = \{1\ 2\}\ 6\ 5\ 4\ 3\ 7$, then $d_b(\Pi, \Gamma) = 2$ while $d_r(\Pi, \Gamma) = 1$.

2.4. Common adjacencies of two genomes

Given two genomes in the total order, we already know that a breakpoint refers to two markers that are adjacent in one genome but not in another. If two markers are adjacent in both genomes, we will say that they form a *common adjacency*. An important observation is that, for either genome, the number of breakpoints (i.e., the breakpoint distance) plus the number of common adjacencies is one less than the size of the genome. As a result, in computing the breakpoint distance between two partially-ordered genomes, finding two respective linearizations (e.g., π and γ in the above definition) that minimize the number of breakpoints is equivalent to finding two linearizations that maximize the number of common adjacencies.

Given two partially-ordered genomes Π and Γ, two markers are said to be a *possible common adjacency* if they could appear as a common adjacency in some pair of respective linearizations of Π and Γ (Fu and Jiang, 2007). Note that all the possible common adjacencies may not coexist in any fixed pair of respective linearizations. For example, there are five possible common adjacencies between $\Pi = \{1\ 2\}\ 4\ \{3\ 5\ 6\}\ 7$ and $\Gamma = 1\ 2\ 3\ 4\ 5\ 6\ 7$; they are $\langle 1, 2\rangle$, $\langle 3, 4\rangle$, $\langle 4, 5\rangle$, $\langle 5, 6\rangle$ and $\langle 6, 7\rangle$. However, the maximum number of common adjacencies that could exist between a pair of two respective linearizations of Π and Γ is only four.

3. METHODOLOGY

Given a set of K individual genetic maps, $\Omega = \{\Pi_1, \Pi_2, \cdots, \Pi_K\}$ and another reference map Γ in the total order, we aim to find a consensus genetic map that retains the order relations of a maximum weighted acyclic subgraph induced from the aggregate graph Ω while minimizing its rearrangement distance to the reference map Γ. To avoid the inclusion of any spurious order, we hence require that all the order relations in the final consensus genetic map be either (i) implied by the aggregate graph, (ii) needed to minimize the rearrangement distance in comparative analysis, or (iii) imposed due to the restrictions on the graphical structure of a consensus genetic map (i.e., a linear order of marker bins). In other words, the consensus genetic map shall not include any spurious order relations without supportive evidence. Formally, the new problem is defined as follows:

Input: A set of K individual genetic maps $\Omega = \{\Pi_1, \Pi_2, \cdots, \Pi_K\}$ and a reference genome Γ in total order.

Output: A consensus genetic map in linear order of marker bins which (i) respects the order relations of a maximum weighted acyclic subgraph of Ω, and (ii) achieves the lowest possible rearrangement distance to the reference genome Γ.

An illustrative example is given in Figure 1, where the genome Ω has two individual genetic maps, $\Pi_1 = 2\ \{5\ 6\}\ 4\ 3$ and $\Pi_2 = 1\ 6\ 4\ \{5\ 7\}$. The aggregate graph of Ω is then constructed by combining the two

individual maps. The directed cycle in Ω between the markers 4 and 5 indicates an order conflict which needs to be resolved. With comparison to a totally-ordered genome Γ which is given as the identity permutation, the rearrangement distance is minimized only when marker 4 is ordered immediately after marker 5 in the final consensus genetic map. In particular, if the breakpoint distance is considered, the consensus genetic map $\Omega_b(\Gamma)$ in Figure 1 will be returned, where all the order relations are supported with evidence. For example, marker 7 can only be ordered after marker 3 because the later marker needs to immediately follow marker 4 for the breakpoint distance to be minimized. On the other hand, marker 1 and marker 2 remain incomparable since both orderings between them lead to the same breakpoint distance between $\Omega_b(\Gamma)$ and Γ. If the reversal distance is instead considered, the order between marker 1 and marker 2 can be resolved in the resulting consensus genetic map $\Omega_r(\Gamma)$ (Fig. 1). In general, $\Omega_r(\Gamma)$ further refines $\Omega_b(\Gamma)$ with more incomparable orders resolved.

Observe from the above example that comparative analysis can not only contribute to inferring new orders, but also to resolving order conflicts among individual genetic maps. It is such a comparative context that distinguishes our present study from all the previous studies on the construction of consensus genetic maps (Jansen et al., 2001; Jackson et al., 2005, 2008; Wu et al., 2008a,b) or on the linearization of partially-ordered genetic maps (Zheng et al., 2005; Zheng and Sankoff, 2006; Blin et al., 2007; Fu and Jiang, 2007; Chen and Cui, 2009).

4. ALGORITHMS

To follow the proposed methodology closely, we adopt an approach that starts with finding all the maximum weighted acyclic subgraphs of Ω. Among these maximum weighted acyclic subgraphs, we then choose the one that attains the minimum rearrangement distance to the reference map Γ and refine it to produce a consensus genetic map (in linear order of marker bins). For example, the genome Ω in Figure 1 has two acyclic subgraphs with the maximum weight; they are $A(\Omega)_1$ and $A(\Omega)_2$ as shown in Figure 2. Their breakpoint distances to the reference map Γ are respectively 1 and 2. Therefore, $A(\Omega)_1$ is chosen rather than $A(\Omega)_2$. In the second refinement step, we place marker 5 between marker 6 and marker 4 and also marker 3 between marker 4 and marker 7, since they are necessary to ensure the minimum breakpoint distance between $A(\Omega)_1$ and Γ. Consequently, $\Omega_b(\Gamma)$ of Figure 1 is returned as the consensus genetic map. If the reversal distance is considered instead of the breakpoint distance, we will further place marker 1 before marker 2 to obtain $\Omega_r(\Gamma)$ of Figure 1 as the final consensus genetic map.

An alternative way to implement the second refinement step is as follows. First, we linearize the selected acyclic subgraph $A(\Omega)_1$ into a total order, and then remove from this total order all the spurious orders (i.e., those orders without any evidence support from the mapping study or comparative study) by merging neighboring marks into bins. For example, if $A(\Omega)_1$ is linearized into $L(A(\Omega)_1)$ by minimizing the breakpoint distance to Γ (Fig. 2), then we shall regard the relative order between marker 1 and marker 2 as spurious. In this case, we will merge them into the same bin.

Note that finding even one maximum weighted acyclic subgraph of Ω is already hard in general, since it is essentially equivalent to the NP-hard problem of *minimum weighted feedback edge set* (Garey and Johnson, 1979). Therefore, we do not expect an efficient algorithm that can construct an optimal consensus genetic map by following our proposed approach. Instead, we formulate below two integer linear programming (ILP) models, one in polynomial size and the other in linear size, in order to take advantage of the existing powerful solvers for ILP problems, such as IBM ILOG CPLEX.

4.1. A polynomial-sized ILP formulation

Let $\{1, 2, \cdots, n\}$ be the marker set of the genome $\Omega = \{\Pi_1, \Pi_2, \cdots, \Pi_K\}$. To simplify the exposition and without loss of generality, we assume that the reference genome Γ is the identity permutation; that is, $\Gamma = 1\ 2\ \cdots\ n$. As such, a common adjacency of Ω and Γ can only be a pair of two markers with consecutive indices, i.e., i and $i+1$, where $1 \le i \le n-1$.

Observe that a maximal subset of markers whose orders are *pairwise* conflicting comprises a *strongly connected component* (SCC) of the directed graph Ω—that is, a maximal subgraph in which each vertex has a directed path to every other vertex. If we shrink every SCC down to a single vertex and draw an arc

between two of them if there is an arc from some vertex in the first to some vertex in the second, the resulting graph would be a directed acyclic graph and hence can be topologically sorted. A number of very efficient algorithms are able to find the SCCs in a directed graph in linear time, such as Tarjan's algorithm (Tarjan, 1972) which is based on depth-first search. We implemented Tarjan's algorithm to facilitate our ILP formulation below.

Before presenting our ILP formulation, we build the following nine subsets comprising of pairs of distinct markers. An ordered pair of distinct markers $\langle i, j\rangle$ is in the subset

- $\mathcal{P}_{11}$ if (a) there exists an arc from marker i to marker j in every individual genetic map Π_i, for all $1 \le i \le K$, and (b) either $j = i + 1$ or $i = j + 1$;
- $\mathcal{P}_{12}$ if (a) there exists an arc from marker i to marker j in every individual genetic map Π_i, for all $1 \le i \le K$, and (b) both i and j are the only makers in their respective bins in every individual genetic map Π_i, for all $1 \le i \le K$;
- $\mathcal{P}_1$ if $\langle i, j\rangle$ is in either the subset $\mathcal{P}_{11}$ or the subset $\mathcal{P}_{12}$ (that is, $\mathcal{P}_1 = \mathcal{P}_{11} \biguplus \mathcal{P}_{12}$);
- $\mathcal{P}_{21}$ if (a) $\langle i, j\rangle$ is not in the subset $\mathcal{P}_1$, and (b) there exists a directed path from marker i to marker j in every individual genetic map Π_i, for all $1 \le i \le K$;
- $\mathcal{P}_{22}$ if (a) $\langle i, j\rangle$ is not in the subset $\mathcal{P}_1$, (b) i and j belong to different SCCs of Ω, and (c) there exists a directed path from marker i to marker j in Ω;
- $\mathcal{P}_2$ if $\langle i, j\rangle$ is in either the subset $\mathcal{P}_{21}$ or the subset $\mathcal{P}_{22}$ (that is, $\mathcal{P}_2 = \mathcal{P}_{21} \biguplus \mathcal{P}_{22}$);
- $\mathcal{P}_3$ if (a) $\langle i, j\rangle$ is not in either of $\mathcal{P}_1$ or $\mathcal{P}_{21}$, (b) i and j belong to the same SCC, and (c) there exists an arc from marker i to marker j in Ω;
- $\mathcal{P}_4$ if (a) $i < j$, and (b) neither $\langle i, j\rangle$ nor $\langle j, i\rangle$ exists in either of $\mathcal{P}_1$ $\mathcal{P}_2$ or $\mathcal{P}_3$;
- $\mathcal{O}$ if (a) $j = i + 1$, (b) $\langle i, j\rangle$ is not in the subset $\mathcal{P}_1$, and (c) there does not exist in Ω a directed path from marker i to marker j or from marker j to marker i which passes through a vertex belonging to a SCC different from the SCCs containing i or j.

Consider the genome Ω given in Figure 1. There is only one SCC containing more than two vertices, and the elements contained in the nine subsets are listed below.

$$
\begin{aligned}
\mathcal{P}_1 &= \mathcal{P}_{11} = \mathcal{P}_{12} = \emptyset,\\
\mathcal{P}_{21} &= \{\langle 6,4\rangle\},\\
\mathcal{P}_{22} &= \{\langle 1,3\rangle, \langle 1,4\rangle, \langle 1,5\rangle, \langle 1,6\rangle, \langle 1,7\rangle, \langle 2,3\rangle, \langle 2,4\rangle, \langle 2,5\rangle, \langle 2,6\rangle,\\
&\qquad \langle 2,7\rangle, \langle 4,3\rangle, \langle 4,7\rangle, \langle 5,3\rangle, \langle 5,7\rangle, \langle 6,3\rangle, \langle 6,4\rangle, \langle 6,5\rangle, \langle 6,7\rangle\},\\
\mathcal{P}_2 &= \mathcal{P}_{22},\\
\mathcal{P}_3 &= \{\langle 4,5\rangle, \langle 5,4\rangle\},\\
\mathcal{P}_4 &= \{\langle 1,2\rangle, \langle 3,7\rangle\},\\
\mathcal{O} &= \{\langle 1,2\rangle, \langle 3,4\rangle, \langle 4,5\rangle, \langle 5,6\rangle\}.
\end{aligned}
$$

Note that $\langle 6, 4\rangle$ is not included into the subset $\mathcal{P}_1$ because it does not satisfy the second condition used to define either $\mathcal{P}_{11}$ or $\mathcal{P}_{12}$.

The marker pairs contained in $\mathcal{P}_{11}$ are adjacent not only in all the individual genetic maps but also in the reference genome Γ. Meanwhile, all the individual maps have identified for each pair a consistent ordering. Therefore, it is desirable to retain both of their adjacencies and order relations in the final consensus genetic map to be constructed. The marker pairs in $\mathcal{P}_{12}$ hold all the above properties except that they may not be adjacent in the reference genome Γ. However, all the markers involved are additionally required to be the only markers in their respective bins, which enables their adjacencies and order relations to be preserved in the final consensus genetic map without losing optimality. For the marker pairs in $\mathcal{P}_2$, since the order relations from $\mathcal{P}_{21}$ are consistent across all the individual genetic maps and those from $\mathcal{P}_{22}$ are not involved in any order conflict, we will retain only their order relations (not necessarily being adjacent) in the final consensus genetic map. The subset $\mathcal{P}_3$ instead consists of those marker pairs with conflicting orders, which need to be resolved. $\mathcal{P}_4$ contains marker pairs unordered by Ω (i.e., there does not exist in Ω a directed path from one marker to another). Lastly, the subset $\mathcal{O}$ consists of all the possible common adjacencies between Ω and Γ. Note that although both $\mathcal{P}_1$ and $\mathcal{P}_{21}$ are likely to contain a pair $\langle i, j\rangle$ of two markers belonging to the same SCC of Ω so that their order is indeed in conflict, we still choose to order i before j in the final

consensus genetic map because this order is already convinced by all the individual genetic maps under consideration.

Further notice that the size of $\mathcal{P}_3$ can never exceed the total number of arcs in Ω. The union $\mathcal{P}_1 \bigcup \mathcal{P}_2 \bigcup \mathcal{P}_3 \bigcup \mathcal{P}_4$ contains at least one ordered pair of either $\langle i, j \rangle$ or $\langle j, i \rangle$, for all the distinct markers i and j. The subset $\mathcal{O}$ might overlap with any of the subsets $\mathcal{P}_2$, $\mathcal{P}_3$ and $\mathcal{P}_4$. As illustrated in the aggregate graph of Ω in Figure 1, each pair $\langle i, j \rangle$ in $\mathcal{P}_3$ is associated with a weight value w_{ij}, which will be used in our ILP formulations below.

Our first ILP formulation, presented on the left of Figure 3, is able to construct an optimal consensus genetic map in total order. In this formulation, x_i is an integer variable defining the relative position of marker i in the final consensus genetic map to be constructed, y_{ij} is a binary variable indicating whether marker i is ordered before marker j, and z_i is also a binary variable but indicating whether marker i and marker $i+1$ are consecutive in the final consensus genetic map. Below, we briefly discuss the constraints and the objective function.

- Constraint (C.1) ensures that the final consensus genetic map retains those common adjacencies convinced by all the input individual genetic maps and by the reference genome as well.
- Constraint (C.2), together with Constraint (C.1), ensures that the final consensus genetic map retains all the orders that are already resolved by the input individual genetic maps.
- Constraints (C.3) and (C.4) ensure that all the unresolved orders, including the order-conflicting pairs in $\mathcal{P}_3$ and the unordered pairs in $\mathcal{P}_4$, are to be resolved in the final consensus genetic map. Note that $y_{ij} = 1$ if and only if marker i is ordered before j.
- Constraints (C.5) and (C.6) ensure that $z_i = 1$ if and only if $\langle i, i+1 \rangle$ is an adjacency pair in the final consensus genetic map. For an adjacency pair $\langle i, i+1 \rangle$, marker i is located either immediately before marker $i+1$ or immediately after marker $i+1$.
- The objective function consists of two sums. Maximizing the first sum finds an acyclic subgraph of Ω with the maximum weight, while maximizing the second sum retains the maximum number of common adjacencies in the resulting consensus genetic map, or equivalently, minimizes the breakpoint distance to the reference genome Γ. The coefficient n used in the first sum ensures the maximum weighted acyclic subgraph to be found, since the second sum would not yield a value greater than n.

Although we have made efforts to reduce the numbers of variables and constraints in the above ILP formulation, there exist $O(n^2)$ variables and $O(n^2)$ constraints in general. It could inadvertently result in a very time-consuming computation. Therefore, this formulation should be intended only for genetic maps of small size.

A POLYNOMIAL-SIZED ILP FORMULATION

Objective :

$$\text{Maximize} \quad \sum_{<i,j> \in \mathcal{P}_3} n \cdot w_{ij} \cdot y_{ij} + \sum_{<i,i+1> \in \mathcal{O}} z_i$$

Constraints :

(C.1) $x_i - x_j = -1, \quad \forall < i, j > \in \mathcal{P}_1;$

(C.2) $x_i - x_j \leq -1, \quad \forall < i, j > \in \mathcal{P}_2;$

(C.3) $x_i - x_j + n \cdot y_{ij} \leq n - 1, \quad \forall < i, j > \in \mathcal{P}_3 \uplus \mathcal{P}_4;$

(C.4) $x_j - x_i - n \cdot y_{ij} \leq -1, \quad \forall < i, j > \in \mathcal{P}_3 \uplus \mathcal{P}_4;$

(C.5) $x_{i+1} - x_i + (n-2) \cdot z_i \leq n - 1, \quad \forall < i, i+1 > \in \mathcal{O};$

(C.6) $x_i - x_{i+1} + (n-2) \cdot z_i \leq n - 1, \quad \forall < i, i+1 > \in \mathcal{O};$

Domains :

(D.1) $1 \leq x_i \leq n, \quad \forall i \in [1, n];$

(D.2) x_i is integer, $\quad \forall i \in [1, n];$

(D.3) y_{ij} is binary, $\quad \forall < i, j > \in \mathcal{P}_3 \uplus \mathcal{P}_4;$

(D.4) z_i is binary, $\quad \forall < i, i+1 > \in \mathcal{O}.$

A LINEAR-SIZED ILP FORMULATION

Objective :

$$\text{Maximize} \quad \sum_{<i,j> \in \mathcal{P}_3} n \cdot w_{ij} \cdot y_{ij} + \sum_{<i,i+1> \in \mathcal{O}} z_i$$

Constraints :

(C.1) $x_i - x_j = -1, \quad \forall < i, j > \in \mathcal{P}_1;$

(C.2) $x_i - x_j \leq -1, \quad \forall < i, j > \in \mathcal{P}_2;$

(C.3) $x_i - x_j + n \cdot y_{ij} \leq n - 1, \quad \forall < i, j > \in \mathcal{P}_3;$

(C.4) $x_j - x_i - n \cdot y_{ij} \leq -1, \quad \forall < i, j > \in \mathcal{P}_3;$

(C.5) $x_{i+1} - x_i + (n-2) \cdot z_i \leq n - 1, \quad \forall < i, i+1 > \in \mathcal{O};$

(C.6) $x_i - x_{i+1} + (n-2) \cdot z_i \leq n - 1, \quad \forall < i, i+1 > \in \mathcal{O};$

Domains :

(D.1) $1 \leq x_i \leq n, \quad \forall i \in [1, n];$

(D.2) x_i is integer, $\quad \forall i \in [1, n];$

(D.3) y_{ij} is binary, $\quad \forall < i, j > \in \mathcal{P}_3;$

(D.4) z_i is binary, $\quad \forall < i, i+1 > \in \mathcal{O}.$

FIG. 3. Two integer linear programming formulations with $O(n^2)$ and $O(n)$ variables, respectively, where n denotes the total number of markers.

4.2. A linear-sized ILP formulation

The primary determinant of computational difficulty for an ILP problem is the number of integer variables rather than the number of constraints. In order to find an approach computationally feasible for genomes of moderately large size (e.g., having about 1000 markers), it is desirable to have an ILP formulation for which the number of variables scales linearly with the number of markers. To this end, we present here an alternative approach which is based on a slightly different objective.

The previous approach aims to find a consensus genetic map at the minimum breakpoint distance to a given reference genome, i.e., minimizing the number of breakpoints, or equivalently, maximizing the number of common adjacencies between a linearization of Ω and the reference genome Γ. The new approach instead aims to find a consensus genetic map (denoted by $\Omega_{b'}(\Gamma)$) that maximizes its total number of *possible* common adjacencies with the reference genome Γ. Note that these possible common adjacencies occur in both the consensus genetic map $\Omega_{b'}(\Gamma)$ and the reference genome Γ, but may not necessarily occur simultaneously in any fixed linearization of $\Omega_{b'}(\Gamma)$.

As the number of markers in each bin of an individual genetic map is usually upper bounded by a constant for a genetic mapping dataset, the aggregate graph Ω hence contains only $O(n)$ arcs such that the size of $\mathcal{P}_3$ is within $O(n)$. With the new approach above we are able to formulate an ILP model that involves only $O(n)$ variables. As seen in Figure 3 on the right, the new formulation involves n variables x_i, $O(n)$ variables y_{ij} and at most $(n-1)$ variables z_i. It differs from the previous formulation mainly in the constraints (C.3) and (C.4), where the subset $\mathcal{P}_4$ is removed. If we solve for the aggregate graph Ω given in Figure 1 with the new ILP formulation, the resulting consensus genetic map would be $\Omega_{b'}(\Gamma)$ as shown in Figure 2. Although the comparison with the reference genome Γ suggests that the marker 5 shall be ordered immediately after the marker 6, $\Omega_{b'}(\Gamma)$ unfortunately did not determine their relative order.

To solve an ILP problem, we employ a high-performance mathematical programming engine IBM ILOG CPLEX 12.1, which is available at www-01.ibm.com/software/integration/optimization/cplex/. It implements a branch-and-bound search with advanced algorithmic features such as cuts and heuristics, and is particularly suitable for solving the ILP problems with *sparse* coefficient constraint matrices; that is, the percentage of variables per constraint that have nonzero coefficients is quite low. Apparently, this kind of sparsity is common to the ILP problems which we formulated above for constructing consensus genetic maps, because each constraint involves at most three variables. For example, given a genome consisting of 1000 markers, the percentage of variables per constraint that have nonzero coefficients is often less than 0.2% (as would be seen in the simulation tests below).

5. EXPERIMENTAL RESULTS

We implemented our algorithm in C++ and carried out performance tests on both simulated data and real biological data. The implemented software, called ILPMap, is freely available at www1.spms.ntu.edu.sg/~chenxin/ILPMap/. In our experiments below, ILPMap was run on a Windows XP desktop PC with a 3.2GHz Intel Pentium processor and 3.5GB RAM, and MergeMap instead through its web server at http://138.23.191.145/mgmap/.

5.1. Simulated data

The purpose of this set of experiments is to assess the effectiveness of our proposed approach on constructing consensus genetic maps under comparative analysis. The simulated data is generated as follows. Start from a genome Γ, which is simply given as an identity permutation of n distinct markers. Perform γ reversals on the genome Γ to obtain the *true* map (in total order) of the second genome Ω. The boundaries of these reversals are uniformly distributed within the range of the genome. To generate an individual genetic map for the genome Ω, we follow a similar procedure previously proposed in (Wu et al., 2008b). It first swaps α randomly chosen adjacent pairs, and then relocate β randomly chosen markers to a random position. The α swaps are intended to mimic local reshuffles while the β relocations are intended to mimic global displacements. They are two types of errors commonly seen in an individual genetic map, and the latter type of errors occur much less frequently

than the former in practice. In our simulation experiments, we generate two individual genetic maps for each genome Ω, and try to construct a consensus genetic map with comparison to the second genome Γ. Therefore, the quadruple $(n, \gamma, \alpha, \beta)$ specifies the parameters used to generate an individual genetic map for the genome Ω.

As we mentioned earlier, the proposed algorithm based on the ILP formulation may inadvertently run in exponential time in the worst case, especially when there are many global displacement errors occurring in individual genetic maps and/or there are a large number of distinct markers. Therefore, we create the simulated datasets for the following ten specifications of parameter values of the quadruple $(n, \gamma, \alpha, \beta)$: (100, 6, 6, 1), (100, 8, 8, 2), (300, 10, 12, 3), (300, 14, 16, 4), (500, 16, 20, 5), (500, 22, 28, 6), (800, 24, 32, 7), (800, 28, 36, 8), (1000, 30, 40, 9), and (1000, 36, 50, 10). Note that, up to 1000 distinct markers can be permitted in a simulated genome, which suffices to model genetic maps of moderately large size. Moreover, ten global displacement errors can occur in an individual genetic map, which means that there can be up to twenty global displacement errors occurring in two randomly generated individual genetic maps. In addition to a large number of local reshuffle errors being permitted, a wide range of genetic maps—from high-quality to medium-quality and/or from small size to moderately large size—can all be reasonably well modelled. For each specification of parameter values, twenty random instances are generated in our experimental tests.

We apply both ILPMap (which implements our linear-sized ILP formulation) and MergeMap to each experimental simulated dataset, and the resulting consensus genetic maps for the genome Ω are compared with its true map to count the number of *erroneous* marker pairs. We call a pair of markers erroneous when their relative order in the consensus genetic map differs from the order in the true map (Wu et al., 2008b). If a pair of markers is placed into the same bin in the consensus genetic map, which means that their relative order is not determined yet, they are also considered as an erroneous pair but with a weight of 0.5 rather than the usual weight of 1. When the consensus genetic map is identical to the true map, the number of erroneous marker pairs is zero. On the contrary, when the consensus map is the reverse of the true map, the number of erroneous markers will be the largest possible, that is, $n(n-1)/2$. For each experiment that is carried out on twenty data instances generated with one specification of parameter values, we collect the number of erroneous marker pairs, and calculate both the mean and standard deviation. The resulting statistics are listed in Table 1.

As can be seen in Table 1, ILPMap performs considerably better than MergeMap in terms of the average number of erroneous marker pairs per experiment. In all the ten experiments, the consensus genetic maps found by ILPMap contain erroneous marker pairs on average from two to four times fewer than those found by MergeMap. To be more specific, ILPMap successfully finds more accurate consensus genetic maps than MergeMap in 193 (96.5%) out of the 200 tested instances.

Table 1. Comparison Between ILPMap and MergeMap for Various Parameter Specifications of $(n, \gamma, \alpha, \beta)$

$(n, \gamma, \alpha, \beta)$	*MergeMap erroneous pairs*	*ILPMap erroneous pairs*	*Running time*	*Variables*	*Constraints*
(100, 6, 6, 1)	32.5 (22.7)	7.2 (14.7)	0.2 (0.0)	157.5 (7.3)	4991.4 (19.7)
(100, 8, 8, 2)	65.5 (31.9)	27.2 (28.6)	0.2 (0.1)	176.2 (8.8)	4967.4 (39.3)
(300, 10, 12, 3)	298.6 (141.5)	72.0 (107.9)	2.5 (0.3)	431.8 (12.4)	44516.4 (196.8)
(300, 14, 16, 4)	355.4 (103.0)	135.0 (134.7)	2.8 (0.4)	471.6 (7.8)	44432.7 (192.6)
(500, 16, 20, 5)	912.1 (212.2)	328.8 (231.8)	12.4 (1.3)	730.1 (10.0)	123414.3 (308.2)
(500, 22, 28, 6)	1025.2 (318.5)	352.6 (305.3)	13.6 (1.7)	795.2 (15.6)	123214.1 (421.2)
(800, 24, 32, 7)	1842.7 (488.7)	758.3 (403.9)	49.4 (3.8)	1155.5 (14.7)	316604.8 (937.8)
(800, 28, 36, 8)	2151.9 (553.4)	1080.8 (561.3)	50.3 (4.1)	1200.3 (17.8)	316001.8 (839.4)
(1000, 30, 40, 9)	2933.6 (643.4)	1606.7 (909.6)	96.3 (5.6)	1448.1 (19.5)	494455.3 (802.0)
(1000, 36, 50, 10)	3485.5 (836.8)	1743.4 (828.2)	100.8 (8.0)	1527.1 (29.0)	493953.2 (1262.8)

The columns under "erroneous pairs" indicate the average number of erroneous marker pairs obtained from twenty independent data sets for each parameter specification. The columns under "running time," "variables," and "constraints" indicate, for ILPMap only, the average running time in seconds, the average number of variables, and the average number of constraints present in the ILP formulations, respectively. The corresponding standard deviations are reported in parentheses.

Although we expect that ILPMAP would run very slow for some datasets, it is surprisingly very efficient in practical use. For the 200 tested data instances, none takes ILPMAP more than 115 seconds to find the optimal consensus genetic map. As can be seen from Table 1, for example, the average running time for the data instances generated with the parameter specification (1000, 36, 50, 10) is about 100.8 seconds only. Another interesting observation is that the standard deviations of the running time are consistently very small relative to their corresponding average values, which shows the superior effectiveness of our ILP formulation. Because we ran MERGEMAP through its web server we are not able to collect its running time for comparison.

It is also evident from Table 1 that the number of variables employed in an ILP formulation scales quite linearly with the total number of markers n, thereby verifying the assumption that we previously made for the linear-sized ILP formulation. However, it is not the case for the number of constraints, which instead seems to grow quadratically with n. Recall that the primary determinant of computational difficulty for an ILP problem is the number of integer variables rather than the number of constraints, which may explain why we do not see the running of ILPMAP become drastically slow as the number of constraints grow quadratically.

Besides the time efficiency, we also find that ILPMAP is memory efficient in these simulation tests. Although our desktop PC that runs ILPMAP has as large as 3.5GB of main memory, the peak memory usage, including the amount of memory consumed by the operating system, is about 0.8GB only.

5.2. Real data

We illustrate the application of our method to a real biological dataset which appeared in a previous study (Zheng et al., 2005). It contains four individual genetic maps taken from the Gramene database (www.gramene.org/), two of which are generated for chromosome 3 of maize and the other two for the chromosome labelled A and LG-03, respectively, of sorghum. There exist two order conflicts between the two genetic maps of sorghum, involving the markers umc5 versus rz244 and cdo920. To obtain a conflict-free genetic map of sorghum, Zheng et al. (2005) took a simple approach by which all the conflicting order relations are removed so that all the involved markers become incomparable. For the validation purpose, we assume the linear order of the sorghum chromosome found by the previous study of Zheng et al. (2005) to be true.

Here we run ILPMAP on the two individual genetic maps of sorghum to resolve order conflicts, for which the totally-ordered chromosome 3 of maize presented in Zheng et al. (2005) was used as the reference genome for comparative analysis (Fig. 4). It is very interesting to see that the resulting consensus genetic map of sorghum is identical to the one obtained in the study of Zheng et al. (2005). Note that the approach used in Zheng et al. (2005) to derive a total order is based on computing the minimum reversal distance between two input partial orders that are assumed to be conflict-free. In comparison, our approach allows order conflicts to be present among the input individual genetic maps, and then can resolve them successfully. Therefore, our proposed approach shows wider applicability than the approach used in Zheng et al. (2005).

We further run MERGEMAP on the above same data set, and obtain a consensus genetic map which is also depicted in Figure 4. Observe that MERGEMAP is not able to resolve the order conflicts correctly as the marker rz995 is wrongly placed after the marker umc5. Moreover, it introduces many incorrect order relations for markers that are actually not involved in any order conflicts among the input individual genetic maps. For instance, it orders the marker csu690 (i.e., marker 12) before the marker bcd738 (i.e., marker 13), but they have a reverse order as determined by the previous study of Zheng et al. (2005) (and by our approach ILPMAP too). Recall that the main distinguishing feature of ILPMAP is its comparative analysis framework. Therefore, the above experimental results indicate that comparative analysis has been very helpful in constructing more accurate consensus genetic maps.

6. CONCLUSION

In this article we presented a comparative approach to constructing consensus genetic maps, which is based not only on marker order relations available in a given set of individual genetic maps of a species but also on marker order relations from a closely related species. It aims to find a consensus genetic map which

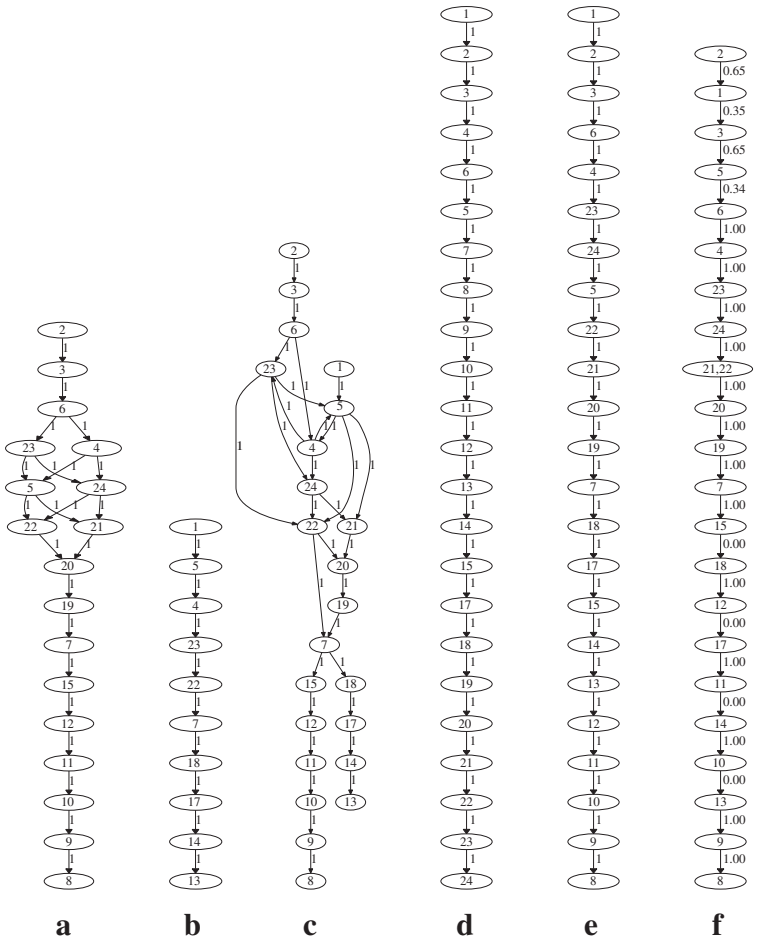

FIG. 4. **(a)** An individual genetic map of sorghum (labeled Paterson 2003). **(b)** The second individual genetic map of sorghum (labeled Klein 2004). **(c)** The combined DG of the two preceding individual genetic maps of sorghum. **(d)** The total order of chromosome 3 of maize presented in Zheng et al. (2005). **(e)** The consensus genetic map of sorghum constructed by ILPMAP. **(f)** The consensus genetic map constructed by MERGEMAP. The corresponding marker names can be found in Zheng et al. (2005), for example, the numbers 4, 5, and 23 denote the markers rz244, umc5, and cdo920, respectively.

retains as many order relations as possible from the input individual genetic maps while minimizing the rearrangement distance to the second reference genome whose markers are already known in total order. To this end, we implemented a polynomial-sized ILP formulation to compute the optimal consensus genetic map, and also a linear-size ILP formulation to compute a (sub-)optimal consensus genetic map at a reduced computational cost. Our preliminary experiments on simulated and real data have demonstrated that our approach performs very well on both resolving order conflicts and linearizing partial orders.

In our experimental studies, ILPMAP was compared to the most recent approach MERGEMAP for the construction of consensus genetic maps. However, the following two factors might have biased the results against MERGEMAP. First, ILPMAP took advantage of a reference genome but MERGEMAP did not. It may make a comparison unfair for MERGEMAP. Therefore, the superiority of ILPMAP over MERGEMAP shall be interpreted only as the utility of comparative analysis in improving the construction of consensus genetic maps. Second, the ground truth is unknown about the true genetic map of sorghum in the real data test. It turns out that the consensus genetic map found by MERGEMAP could still be true, although it is neither consistent with the one previously constructed in the study of Zheng et al. (2005) nor evidenced by a phylogenetically-related genome under comparative analysis.

Comparative analysis is known to provide profound insights into the genome structure and organization. To make such an analysis successful, an appropriate choice of the model organism as reference plays a very important role. The general principle is to choose one that is phylogenetically as close as possible to the organism under investigation. To apply this principle, our comparative approach to constructing consensus genetic maps shall not be an exception.

The major limitation of our ILP-based algorithm is its exponential running time in the worst case, although this is not observed in our simulation tests. Therefore, ILPMAP is well suited for solving

problems of small or medium sizes only. In fact, such a limitation is commonly expected for any algorithm attempting to solve an NP-hard problem exactly, unless P = NP. To overcome this limitation, we may employ a heuristic procedure, probably in the same way as in all the previous studies (Jackson et al., 2005, 2008; Wu et al., 2008b), to break a problem of large size into several subproblems of small size such that every subproblem can be solved with our ILP-based algorithm in an acceptable amount of time. In the future, we plan to look into developing an efficient (e.g., approximate or fixed-parameter tractable) algorithm to construct very accurate consensus genetic maps under comparative analysis.

ACKNOWLEDGMENTS

We would like to thank Dr. Chek Beng Chua for his help in language correction and the anonymous referees for their constructive suggestions and criticism, which greatly improved the manuscript. This work was partially supported by the Singapore NRF grant NRF2007IDM-IDM002-010 and MOE AcRF Tier 1 grant RG78/08.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Bertrand, D., Blanchette, M., and El-Mabrouk, N. 2008. A phylogenetic approach to genetic map refinement. *Proc. RECOMB-CG* 198–210.

Bertrand, D., Blanchette, M., and El-Mabrouk, N. 2009. Genetic map refinement using a comparative genomic approach. *J. Comput. Biol.* 16, 1475–1486.

Blin, G., Blais, E., Hermelin, D., et al. 2007. Gene maps linearization using genomic rearrangement distances. *J. Comput. Biol.* 14, 394–407.

Chen, X., and Cui, Y. 2009. An approximation algorithm for the minimum breakpoint linearization problem. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 6, 401–409.

Fu, Z., and Jiang, T. 2007. Computing the breakpoint distance between partially ordered genomes. *J. Bioinform. Comput. Biol.* 5, 1087–1101.

Garey, M. and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., New York.

Jackson, B., Aluru, S., and Schnable, P. 2005. Consensus genetic maps: a graph theoretic approach. *Proc. IEEE Comput. Syst. Bioinform. Conf.* 35–43.

Jackson, B., Schnable, P., and Aluru, S. 2008. Consensus genetic maps as median orders from inconsistent sources. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 5, 161–171.

Jansen, J. de Jong, A.G., and van Ooijen, J.W. 2001. Constructing dense genetic linkage maps. *Theoret. Appl. Genet.* 102, 1113–1122.

Stam, P. 1993. Construction of integrated genetic linkage maps by means of a new computer package: Joinmap. *Plant J.* 3, 739–744.

Tarjan, R. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.* 1, 146–160.

Wu, Y., Bhat, P., Close, T., et al. 2008a. Efficient and accurate construction of genetic linkage maps from the minimum spanning tree of a graph. *PLoS Genet*. 4, e1000212.

Wu, Y., Close, T., and Lonardi, S. 2008b. On the accurate construction of consensus genetic maps. *Proc. LSS Comput. Syst. Bioinform. Conf.* 285–296.

Yap, I., Schneider, D., Kleinberg, J., et al. 2003. A graph-theoretic approach to comparing and integrating genetic, physical and sequence-based maps. *Genetics* 165, 2235–2247.

Zheng, C., and Sankoff, D. 2006. Genome rearrangements with partially ordered chromosomes. *J. Combin. Optimiz.* 11, 133–144.

Zheng, C., Lenert, A., and Sankoff, D. 2005. Reversal distance for partially ordered genomes. *Bioinformatics* 21, i502–i508.

Zheng, C., Zhu, Q., and Sankoff, D. 2007. Removing noise and ambiguities from comparative maps in rearrangement analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 4, 515–522.

Address correspondence to:
Dr. Xin Chen
Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore

E-mail: chenxin@ntu.edu.sg