Consensus Genetic Maps as Median Orders from Inconsistent Sources

Benjamin N. Jackson, Patrick S. Schnable, and Srinivas Aluru

Abstract—A genetic map is an ordering of genetic markers calculated from a population of known lineage. Although, traditionally, a map has been generated from a single population for each species, recently, researchers have created maps from multiple populations. In the face of these new data, we address the need to find a consensus map—a map that combines the information from multiple partial and possibly inconsistent input maps. We model each input map as a partial order and formulate the consensus problem as finding a median partial order. Finding the median of multiple total orders (preferences or rankings) is a well-studied problem in social choice. We choose to find the median by using the weighted symmetric difference distance, which is a more general version of both the symmetric difference distance and the Kemeny distance. Finding a median order using this distance is NP-hard. We show that, for our chosen weight assignment, a median order satisfies the positive responsiveness, extended Condorcet, and unanimity criteria. Our solution involves finding the maximum acyclic subgraph of a weighted directed graph. We present a method that dynamically switches between an exact branch and bound algorithm and a heuristic algorithm and show that, for real data from closely related organisms, an exact median can often be found. We present experimental results by using seven populations of the crop plant *Zea mays*.

Index Terms—Genetic map, median order, path and circuit problems, Kemeny distance, symmetric difference distance.

1 Introduction

Constructing a genetic map is an involved process which has traditionally led researchers to focus on creating a single reference map from a single population for each organism. However, to order a mapping locus, one requires polymorphisms at that locus within the mapping population. Because no population can contain polymorphisms for all of the desired mapping loci, it is useful to construct maps from multiple populations. This trend has identified the need to construct a consensus reference map —a map that combines mapping data from multiple population studies.

We address the problem of creating such a map, with the caveat that the population maps might be inconsistent. We focus on inconsistencies that can arise through errors in the map making process. As it is harder to properly evaluate the ordering of two proximate markers versus two distant markers, this can be thought of as a resolution problem. Although map making methods strive to eliminate errors, in practice, misorderings can occur. In addition to experimental error, it is possible that one population carries a rearrangement relative to the other populations. At some level, it is impossible to analytically discern between the possibilities of actual differences and experimental errors. However, given that experimental errors are more likely to be observed than actual rearrangements, it is reasonable to assume that any minor inconsistencies observed are due to errors.

We relate the consensus finding problem to similar problems first addressed in the study of social choice. Specifically, we formulate this problem as finding a median relation between multiple binary relations under the weighted symmetric difference distance, which is a more general version of the well-known symmetric difference distance [17] and Kemeny distance [20], and we note that finding a median by using this distance is NP-hard.

Finding a median sequence or order has also been applied to the problem of reconstructing the ancestral genome in phylogenetic studies. In this case, the distance function is motivated by the evolutionary model [28]. A well-studied distance for this purpose is the reversal distance, where the edit operations are signed reversal and, perhaps, translocation [15], [16], [38]. Algorithms for finding the reversal distance were originally designed for total orders, although, recently, researchers have adopted these algorithms to work with partial orders [31].

We model a genetic map as a partial order. Modeling the input as a partial order instead of a linear order allows for flexibility in specifying the input maps. To represent the partial order, we use directed acyclic graphs (DAGs). Nodes in a graph correspond to markers in a map, while paths correspond to the ordering information. As we consider the input maps to be imperfect observations of the same phenomenon, our primary focus is to discover the most likely underlying order, given the inputs. Thus, the method is quite different from that described in [40], which also represents maps as DAGs but focuses on data aggregation.

In finding the consensus order, we rely on many graph algorithms, including all-pair shortest paths, transitive closure and reduction, simple-cycle enumeration, minimum-feedback arc set, and strongly connected

- *B.N. Jackson and S. Aluru are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50014. E-mail: {zbbrox, aluru}@iastate.edu.*
- *P.S. Schnable is with the Center for Plant Genomics, Iowa State University, 2035B Carver CO-LAB, Ames, IA 50011-3650. E-mail: schnable@iastate.edu.*

Manuscript received 7 July 2006; revised 12 Feb. 2007; accepted 13 May 2007; published online 8 June 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0135-0706.
Digital Object Identifier no. 10.1109/TCBB.2007.70221.

component enumeration. All of these problems are well studied in graph theory and the most limiting is finding the minimum-feedback arc set because it is NP-hard. There is an $O(\log n \log\log n)$ approximation algorithm for the minimum-feedback arc set [10] and other solution methods are presented in [6], [9]. We present a heuristic algorithm that we use in combination with an exact solver, both of which use a set cover formulation of the problem. The problem can often be solved exactly in practice, demonstrated by the experimental results presented in Section 9.

2 Error in Constructing Genetic Maps

Humans and most other plant and animal species contain two homologous copies of each chromosome. Each homologous chromosome contains a copy (allele) of each gene. These alleles can be either the same or different for a specific gene and we correspondingly term the organism as either homozygous or heterozygous for that gene. One can extend this idea to a population of individuals, terming the population monomorphic or polymorphic for that gene. During the sexual reproduction of a diploid, we can think of a daughter chromosome as constructed by copying information from either homologous parent chromosome. The points at which the copying switches from one parent to the other are known as crossovers. Thus, the genetic material in the daughter is a recombination of the material in the parents. Given the genotypes of the parent and the progeny and polymorphisms at the locations of interest, one can deduce if the alleles of two genes along the daughter came from the same or different chromosomes of the parent.

The frequency at which alleles of two genes came from different chromosomes of the parent is called the recombination fraction for that pair of genes. Because of the crossover mechanism, two close genes will have a small recombination fraction, while two distant genes will have a fraction of approximately $\frac{1}{2}$ as an odd or even number of crossovers is equally likely to occur between them. Thus, the recombination fraction can be converted to a distance of sorts, although this distance is not considered a distance metric. The reader is referred to [30] for more detail. Regardless of the exact nature of map construction—be it a maximum likelihood [21], [22], [30], [32] or a traveling salesman-based method [11], [27], [26], [29]—maps are prone to local errors due to the nature of the distance data. Although a marker can be placed fairly certainly in the proper neighborhood, even if we make the best placement given the data, we cannot have perfect confidence in the accuracy of the maps. Statistical methods such as bootstrapping are used to minimize the error, but they cannot eliminate it.

3 Graph Representations of Partial Orders

We model a genetic map M_i as a partial order $\prec_i$ on a set of markers S_i. We say that $a \prec_i b$ for $\{a, b\} \in S_i$ if and only if a occurs before b in the known order of the markers, determined through the analysis of experimental data. We use the notation $a \overline{\prec}_i b$ as equivalent to $\neg(a \prec_i b)$. We use the notation $a \|_i b$ to indicate that a and b are incomparable in $\prec_i$, which is equivalent to saying that $(a \overline{\prec}_i b) \wedge (b \overline{\prec}_i a)$.

To manipulate a partial order algorithmically, we represent it as a DAG $G = (V, E)$. We use the notation $\langle u, \ldots, v\rangle$ to denote a path between nodes u and v in G. We define a set G_i of equivalent DAGs that correspond to the order $\prec_i$.

Definition 3.1. *The set G_i is the set of graphs corresponding to the partial order $\prec_i$. $G \in G_i$ if and only if*

1. *there is a bijection between vertices in G and markers in S_i and*
2. *a path $\langle u, \ldots, v\rangle$ exists in G if and only if $u \prec_i v$.*

We will specifically work with two unique graphs in G_i: the transitive reduction and transitive closure graphs [14].

Definition 3.2. *The* **transitive reduction graph** *$G_i^R = (V_i^R, E_i^R) \in G_i$ is the graph such that $(u, k) \in E_i^R \wedge (k, v) \in E_i^R \Rightarrow (u, v) \notin E_i^R$.*

Observation 3.3. *G_i^R is the graph in G_i with the minimum cardinality edge set.*

Definition 3.4. *The* **transitive closure graph** *$G_i^C = (V_i^C, E_i^C) \in G_i$ is the graph such that $(u, v) \in E_i^C \Leftrightarrow u \prec_i v$.*

Observation 3.5. *G_i^C is the graph in G_i with the maximum cardinality edge set.*

The transitive reduction graph is of interest because, as the graph with the least number of edges that still encodes the partial order, it will be the most useful when displaying the partial order. The transitive closure graph is of interest because, with it, we can answer the question, "Is a related to b?" in constant time by looking for the existence of an edge.

4 Kemeny Distance for Comparing Orders

In the consensus map problem, we have multiple imperfect orderings, each being an approximation of the same underlying order. The need to find the consensus among multiple rankings or orders was identified as a problem by social scientists interested in studying democratic processes in the late 18th century. Modern study of the problem is attributed to Kemeny [20], who described the problem of finding a median relation, given a set of input relations. Researchers often use how well a particular aggregation scheme approximates the Kemeny median to evaluate that scheme's goodness [36].

Definition 4.1. *The* **Kemeny distance** *between two total orders, $<_i$ and $<_j$, denoted $K(<_i, <_j)$, is a count of the number of pairwise conflicts between the orders. A pair (s_u, s_v) is conflicting if $s_u <_i s_v$ and $s_v <_j s_u$.*

Definition 4.2. *Given a set of total orders $\{<_1, \ldots, <_k\}$, the* **Kemeny median** *$<_M$ is an order that minimizes the function $\sum_{i=1}^{k} K(<_M, <_i)$.*

Given a probability assignment for flipping two adjacent elements, we can calculate the joint probability of one order arising from an antecedent order, assuming that it is constructed using a sequence of such steps. Under this probability assignment, a median order that optimizes the

Kemeny distance is also an order that most likely produces the set of input orders as it is the order for which the least number of moves is required in aggregate [8].

Let $\{\prec_1, \prec_2, \ldots, \prec_k\}$ be a set of partial orders. Let $\prec_A$ be the order created using some order aggregation scheme. The Kemeny median is an aggregate order that satisfies the following criteria:

Definition 4.3. *The* **positive responsiveness criterion** *[3] asks that if $u\|_A v$, then changing $v \prec_i u$ to $u \prec_i v$ for some $\prec_i$ or changing $u\|_i v$ to $u \prec_i v$ for some $\prec_i$ will result in $u \prec_A v$.*

Definition 4.4. *The* **Condorcet paradox** *is the term given to the idea that a majority of orders might have $u \prec v$, $v \prec w$, and $w \prec u$. In this case, the consensus order cannot contain all orderings indicated by a majority of voters.*

Definition 4.5. *The* **Condorcet criterion** *[24] asks that if there exists u such that, for all $v, u \prec_i v$ in the majority, then, for all $v, u \prec_A v$. In other words, u is the winner.*

Definition 4.6. *The* **extended Condorcet criterion** *[35] asks that if a pair $\{u, v\}$ is not involved in a cycle as identified by the Condorcet paradox, $u \prec_i v$ for a majority of inputs implies that $u \prec_A v$.*

Definition 4.7. *The* **unanimity criterion** *asks that if all orders rank $u \prec_i v$, then $u \prec_A v$.*

Finding a median under the Kemeny distance is NP-hard [37] and a $\frac{4}{3}$rd approximate algorithm exists which reduces the problem to finding a minimum-feedback arc set in tournaments [2].

5 Median Partial Orders and the Weighted Symmetric Difference Distance

Some of the notations used in this section has been adapted from [4], [17].

Definition 5.1. *A* **profile** *$\Pi = \{\prec_1, \ldots, \prec_k\}$ is a set of partial orders. Each order $\prec_i$ has a corresponding element set S_i.*

We wish to assign weights corresponding to our confidence in each ordering. We denote the weight assigned to an ordered pair $u \prec_i v$ as $w_i(u, v)$.

Definition 5.2. *A weight assignment w_i is* **nondecreasing** *if the following property holds: $\forall\{t, u, v\}$, $(t \prec_i u) \wedge (u \prec_i v) \Rightarrow (w_i(t, v) \geq w_i(t, u)) \wedge (w_i(t, v) \geq w_i(u, v))$.*

Definition 5.3. *A weight assignment w_i is* **symmetric** *if $\forall\{u, v\}$, $w_i(u, v) = w_i(v, u)$.*

Definition 5.4. *A weight assignment w_i is* **positive** *if $\forall\{u, v\}$, $u \prec_i v \Rightarrow w_i(u, v) > 0$.*

Definition 5.5. *The* **weighted symmetric difference distance** *between two partial orders $\prec_i$ and $\prec_j$, denoted $\delta(\prec_i, \prec_j)$, is a summation over all pairs $u, v \in (S_i \cup S_j)$, as described in Fig. 1.*

Observation 5.6. *If $u \prec_i v$ and $v \prec_j u$ and w_i and w_j are symmetric, then a total of $2(w_i(u, v) + w_j(u, v))$ will be added to the distance.*

$$\delta(\prec_i, \prec_j) = \sum_{u,v \in (S_i \cup S_j)} \begin{cases} w_i(u,v) + w_j(u,v) & \text{if } (u \prec_i v) \wedge (u \overline{\prec}_j v) \\ w_i(u,v) + w_j(u,v) & \text{if } (u \prec_j v) \wedge (u \overline{\prec}_i v) \\ 0 & \text{otherwise} \end{cases}$$

Fig. 1. The weighted symmetric difference distance between two partial orders $\prec_i$ and $\prec_j$.

Observation 5.7. *If u and v are ordered in one order but not the other and w_i and w_j are symmetric, then a total of $w_i(u, v) + w_j(u, v)$ will be added to the distance.*

Observation 5.8. *If $\prec_i$ and $\prec_j$ are total orders and for all (u, v), $w_i(u, v) = \frac{1}{4}$, then the weighted symmetric difference distance is exactly the Kemeny distance.*

Definition 5.9. *The* **aggregate distance** *between a partial order $\prec_p$ and a profile Π is the sum of the distances between $\prec_p$ and all orders in Π:*

$$\Delta(\prec_p, \Pi) = \sum_i \delta(\prec_p, \prec_i).$$

Definition 5.10. *A* **median partial order**, *denoted $\prec_M$, is an order with element set $S_M = S_1 \cup S_2 \cup \ldots \cup S_k$ that minimizes $\Delta(\prec_M, \Pi)$. There could be multiple partial orders that minimize this function.*

Computing a median partial order using the symmetric distance between relations is NP-hard [17]. The symmetric difference distance is a special case of the weighted symmetric distance, where all weights are $\frac{1}{2}$. Therefore, computing a median partial order using the weighted symmetric distance is also NP-hard.

Finding a median is trivial if the input orders have no conflicts, that is, $\nexists u, v, i, j$ such that $u \prec_i v$ and $v \prec_j u$. In this case, a median can be found by taking the superposition of the DAGs representing the orders. The resulting DAG corresponds to a median. However, the problem is more difficult in general, where one might have $u \prec_i v$ and $v \prec_j u$. In this case, we must find a median using a more complicated method.

In our application, we define the weight $w_i(u, v) = w_i(v, u)$ as the length of the shortest path from u to v or v to u in G_i^R. We use the classic $O(n^3)$ Floyd-Warshal algorithm [12] to find the pairwise shortest paths. If no path exists between u and v or v and u, then we assign $w_i(u, v) = w_i(v, u) = 0$. This weight assignment is nondecreasing, symmetric, and positive. Our choice of weight functions is supported by our previous observation that, within a genetic map, local mistakes in order are much more likely than global mistakes.

For the median relation, we choose $w_M(u, v) = 0$ for all u and v. This assignment is also nondecreasing and symmetric. Even if it were obvious what nonzero weight to assign to the pairs in $\prec_M$, if we were to alternately choose $w_M(u, v) > 0$, we would penalize any ordering in the median $u \prec_M v$ if $u\|_i v$ for some $\prec_i \in \Pi$. This contradicts the goal of having the median contain as much ordering information as consistent with Π.

We will show that, under our chosen weight function, a few nice properties hold. First, u and v can be unordered in $\prec_M$ only if the total weight in Π supporting $u \prec v$ is equal

to the total weight in Π supporting $v \prec u$. Second, the median satisfies the positive responsiveness, extended Condorcet, and unanimity criteria defined in Section 4.

We will make use of the unique transitive reduction G_i^R and closure G_i^C graphs for each input relation $\prec_i$, as described in Section 3. We use the conceptual inverse of the Floyd all-pair shortest paths algorithm to find the transitive reduction [1]. The details are left to the reader, with a good reference being [5]. The interested reader is referred to [7] for a recent treatment on closure and its uses.

Consider the global set of markers $S_M = S_1 \cup S_2 \cup \ldots \cup S_n$. Let t be the number of markers in S_M. We represent the graph G_i^C using a $t \times t$ matrix M_i such that $M_i[u,v] = w_i(u,v)$ and $M_i[v,u] = -w_i(u,v)$ for all $u \prec_i v$.

Definition 5.11. *The* **aggregate matrix** $M_A = \sum_{i=1}^{n} M_i$.

Definition 5.12. *The* **aggregate graph** $G_A = (V_A, E_A)$ *has* $V_A = S_M$, *with* $(u,v) \in E_A \Leftrightarrow M_A(u,v) > 0$. *The edge weight of each edge is* $\omega_A(u,v) = M_A[u,v]$.

Definition 5.13. *The* **aggregate relation** R *is the binary relation corresponding to the aggregate graph.* uRv *if and only if a path* $\langle u, \ldots, v \rangle$ *exists in* G_A.

Definition 5.14. *The* **aggregate weight** *for an ordered pair* (u,v), *denoted* $W(u,v)$, *is the sum of the weights* $w_i(u,v)$ *for all* $\prec_i$, *with* $u \prec_i v$.

Observation 5.15. *By construction,* $M_A[u,v] = \omega_A(u,v) = W(u,v) - W(v,u)$.

M_A can be thought of as a weighted vote for each possible ordering $u \prec v$ or $v \prec u$. After tallying the votes of each map, the resulting relation R may be asymmetric if G_A contains cycles. However, let us first consider the case in which G_A contains no cycles, in which case R is a partial order. In this case, R is a median partial order.

Theorem 1. *Assume that* R *is a partial order. Then,* R *is a median.*

Proof. Let u, $v \in S_M$. There are three cases:

Case 1. uRv. In this case, the aggregate distance is increased by $2W(v,u)$ because of this pair. $uRv \Rightarrow M_A[u,v] \geq 0 \Rightarrow 2W(v,u) \leq 2W(u,v)$. Therefore, we cannot decrease the aggregate distance by reordering to vRu. $2W(v,u) \leq 2W(u,v) \Rightarrow 2W(v,u) \leq W(v,u) + W(u,v)$. Therefore, we cannot decrease the aggregate distance by removing the order uRv.

Case 2. vRu. In this case, the aggregate distance is increased by $2W(u,v)$ because of this pair. We cannot reorder u and v or make u and v unrelated and decrease the aggregate distance by the same reasoning as in Case 1.

Case 3. u is not related to v. In this case, the aggregate distance is increased by $W(u,v) + W(v,u)$ because of this pair. Because they are unordered, $M_A[u,v] = M_A[v,u] = 0 \Rightarrow 2W(u,v) = 2W(v,u) = W(u,v) + W(v,u)$. Therefore, by reordering u and v, the aggregate distance will remain the same.

Therefore, for all u, v, changing the ordering of u and v in R will not cause the aggregate distance to decrease and we conclude that R is a median. □

Corollary 5.16. *If two markers* u *and* v *are unordered in* R, *then it must be the case that* $2W(u,v) = 2W(v,u) = W(u,v) + W(v,u)$.

Corollary 5.17. *If two markers* u *and* v *are unordered in* R, *we can create a new relation* R' *by arbitrarily choosing* $uR'v$ *or* $vR'u$. *We can also consider the transitive closure of* R', R'', *which will be a partial order. We have* $\Delta(R'', \Pi) = \Delta(R', \Pi) = \Delta(R, \Pi)$.

6 Median as a Maximum Acyclic Subgraph

In the case where R is not asymmetric, G_A contains cycles that must be broken. This situation arises because of the Condorcet paradox.

Definition 6.1. *Given a weighted cyclic directed graph* $G = (U, V)$, *find an edge set* $E_R = \{(s_1, e_1), \ldots, (s_k, e_k)\}$ *such that* $G' = \{U, V - E_R\}$ *is acyclic and* $\sum_i w(s_i, e_i)$ *is minimized.* E_R *is known as a* **minimum feedback arc set**. G' *is known as a* **maximum acyclic subgraph**.

Observation 6.2. *For each edge* $(u,v) \in E_R$, *the path* $\langle v, \ldots, u \rangle$ *exists in* G'.

Theorem 2. *Let* G_C *be a maximum acyclic subgraph of* G_A. *The relation* $\prec_C$ *corresponding to* G_C *is a median partial order.*

Proof. Let R' be the binary relation such that $uR'v \Leftrightarrow M_A[u,v] > 0$. Let the weight $w_{R'}(u,v) = 0$ for all (u,v). Due to the reasoning presented in Theorem 1, $\Delta(R', \Pi)$ is the minimum aggregate distance between any binary relation and Π. We call this aggregate distance P. P is a lower bound on $\Delta(\prec_M, \Pi)$. If G_A is acyclic, R' is asymmetric, and, because of Corollary 5.16, $P = \Delta(\prec_R, \Pi) = \Delta(\prec_M, \Pi)$. However, G_A might be cyclic. By Observation 5.15 in conjunction with Observation 6.2, we see that, for each edge that we remove in constructing G_C, we are adding the weight $M_A[u,v]$ to the aggregate distance. Therefore, $\Delta(\prec_c, \Pi) = P + \sum_{u,v \in E_R}(M_A[u,v])$. By definition, G_C is that graph, with $\sum_{u,v \in E_R}(M[u,v])$ minimized. Therefore, $\Delta(\prec_C, \Pi)$ is the minimum and we conclude that $\prec_C$ is a median. □

We now can outline the complete method for computing a median, shown in Fig. 2. First, the transitive closures of the inputs are computed. Then, the graphs are merged to form an aggregate graph. Next, cycles are broken in the aggregate. Finally, the transitive reduction is taken and displayed to the user. Although there are multiple orders that minimize the median function, we will from now on refer to a median found using this method as the median.

Theorem 3. *If* w_i *is nondecreasing and positive for all* $\prec_i \in \Pi$ *and we construct* w_M *such that* $w_M(u,v) = 0$ *for all* $u, v \in S_i$, *then the median order* $\prec_M$ *satisfies the unanimity criterion defined in Section 4.*

Proof. Let a and b be elements such that, for all $\prec_i \in \Pi$, $a \prec_i b$. Because $W(a,b) > W(b,a)$, aRb in the aggregate relation R. Therefore, in order for $a \overline{\prec}_M b$ in a median, a and b must be on some cycle in R. For this to be true, there must exist some c such that bRc and cRa. We will show that such a c cannot exist because bRc and cRa are mutually exclusive.

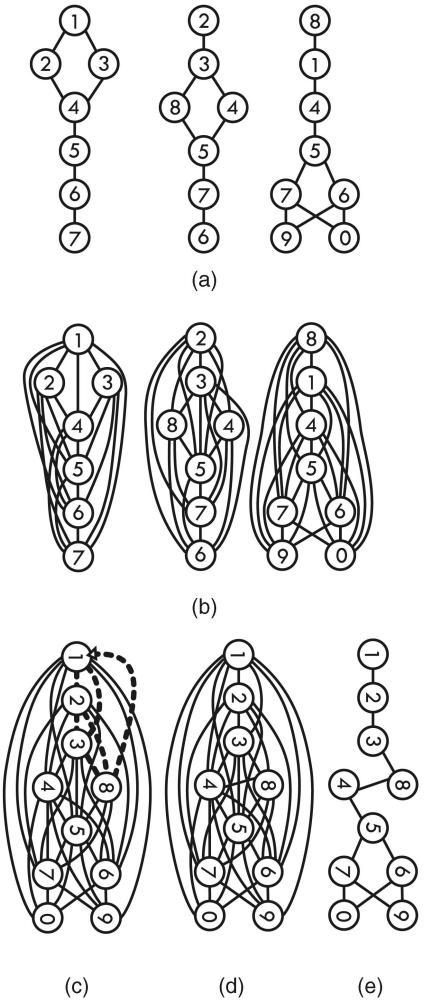

Fig. 2. (a) An example set of input graphs corresponding to a profile Π. All edges are directed in a downward manner. (b) The transitive closure of the input graphs. (c) The aggregate graph G_A. The cyclic subgaph is shown with dotted lines. The single edge moving upward is marked. (d) The maximum acyclic subgraph, which corresponds to $\prec_M$, with the upward traveling edge removed from the aggregate removed. (e) The transitive reduction representation of $\prec_M$.

Choose some element c, $c \neq a$, $c \neq b$. There are five possible ways that c can be ordered relative to a and b in an input order, as shown in Fig. 3. The figure also labels edges with the sum of all pairwise weights for inputs that follow each case. For example, if inputs i and j are exactly those input partial orders that follow Case 1, then $m = w_i(b,c) + w_j(b,c)$.

Assume cRa. Using the edge labels described in Fig. 3, this implies that $y > p + z + v$. We have $q + o + t \geq y + o + t$ due to a nondecreasing weight assignment and $y + o + t \geq y$ due to a positive weight assignment. In addition, $y > p + z + v \geq p$, again, due to a positive weight assignment. Finally, $p \geq m$ due to a nondecreasing weight assignment. Therefore, we have $q + o + t > m$ and we conclude that cRb and $\neg(bRc)$.

Assume bRc. This implies that $m > q + o + t$. By the same reasoning as above, we have $p + z + v \geq m + z + v \geq m > q + o + t \geq q \geq y$. Therefore, we conclude that aRc and $\neg(cRa)$.

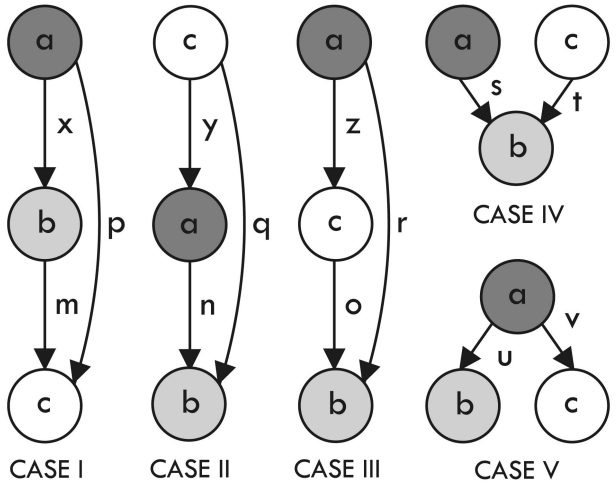

Fig. 3. The possible ways in which an element c can be ordered relative to a and b if $a \prec_i b$ for all $\prec_i \in \Pi$. The labels on the edges represent the sum of the weights for all inputs in the profile that follow that case. For example, if inputs $\prec_i$ and $\prec_j$ followed Case 1, then $m = w_i(b,c) + w_j(b,c)$.

Therefore, $\forall c$, $cRa \Rightarrow \neg(bRc)$, and $bRc \Rightarrow \neg(cRa)$. We conclude that a and b do not lie on a cycle and $a \prec_M b$.□

Theorem 4. *If w_i is symmetric and positive for all $\prec_i \in \Pi$, then the median satisfies the positive responsiveness criterion.*

Proof. Let u and v be some nodes such that $u \|_M v$. Let $\prec_i \in \Pi$ be a partial ordering with either $v \prec_i u$ or $u \|_i v$. We wish to show that if we change the order of u and v such that $u \prec_i v$, then $u \prec_M v$.

Let Π' denote the input profile with this change and W' denote the aggregate weights associated with Π'. Then, $W'(u,v) = W(u,v) + w_i(u,v)$ and $W'(v,u) = W(v,u) - w_i(u,v)$. By Corollary 5.16, $u \|_M v$ only if $W(u,v) = W(v,u)$. Therefore, $W'(u,v) > W'(v,u)$ because w_i is positive. As u and v are unordered in $\prec_M$, having $u \prec'_M v$ will not create a cycle. In addition, ordering $u \prec'_M v$ minimizes the aggregate distance due to the pair (u,v) because $W'(u,v) > W'(v,u)$.

However, for $\prec'_M$ to be a partial order, we must also order $u \prec'_M w$ for all w such that $v \prec_M w$ and $u \|_M w$. By Corollary 5.16, $u \|_M w$ only if $W(u,w) = W(w,u)$. Therefore, if the weight function is symmetric, we can have $u \prec_M w$ without increasing the distance to the median. Hence, we minimize the aggregate distance by ordering $u \prec'_M v$. We conclude that the median using the weighted symmetric difference distance satisfies positive responsiveness. □

Theorem 5. *If we generalize the extended Condorcet criterion to allow for weighted voting, then the median satisfies the extended Condorcet criterion.*

Proof. By construction, the aggregate graph G_A contains the edge (u,v) if and only if a weighted majority of maps has the order (u,v). Let (u,v) be some edge in G_A that is not a part of a cycle. It is easy to see that (u,v) will not be in the minimum feedback arc set. This implies that (u,v) will also be in any median relation. Therefore, finding the

median using the weighted symmetric difference distance satisfies the extended Condorcet criterion. □

7 Minimum Feedback Arc Set as Set Cover

Definition 7.1. *Given a set of elements $U = \{e_1, e_2, \ldots e_n\}$, a* **covering set** *is a set of sets of elements from U, $S = \{s_1, s_2, \ldots s_k\}$ such that $s_1 \cup s_2 \cup \ldots \cup s_k = U$.*

Definition 7.2. *Given a set U, a covering set S, and a weight assignment $w(s_i)$, a* **minimum covering set** *is a covering set $S' \subseteq S$ such that $\sum_{s_i \in S'} w(s_i)$ is minimized. Finding a minimum covering set is the optimization version of the* set cover problem.

Definition 7.3. *Let $C_A = \{c_1, c_2, \ldots, c_f\}$ be the set of all simple cycles in G_A. The* **cycle set** *$C(u, v)$ for edge (u, v) is the set of all simple cycles in G_A that contain the edge (u, v).*

Observation 7.4. *The* **minimum feedback arc set** *E_R is exactly that set of edges such that the corresponding set S, $C(u, v) \in S \Leftrightarrow (u, v) \in E_R$, is the minimum covering set of C_A.*

The set cover problem is NP-hard in the size of S. The greedy approximation algorithm has great success however. The greedy algorithm for the unweighted version of the problem asks that, in each step, you choose the set that covers the most uncovered elements. The greedy approximation has been proven to be within $O(\ln d)$ of the optimal, where d is size of the largest set in S [25]. It has also been shown to be within $O(\ln U \ln \ln U)$ of the optimal in general [33].

A problem that we face in our set cover formulation of the feedback arc set problem is that the elements of U are simple cycles in the graph, which can be exponential in the size of the graph. If we were to instead consider all binary strings of length E, where the assignment of one corresponds to selecting the edge for inclusion in the minimum feedback arc set, we could use a branch-and-bound technique to the search among all 2^E such binary strings for the best solution. In this formulation, we would not have to contend with the possibility of an exponential number of simple cycles. However, the best known cycle enumeration algorithm is optimal, running in $O(C(|V| + |E|))$ time, where C is the number of simple cycles in the graph, as described in [19], and, in practice, there is no problem with enumerating the cycles for the problems that we have looked at.

Formulating the problem as a set cover is attractive, because we can make use of data reduction rules. Iteratively applying the following data reduction rules to the set cover formulation of the problem often makes the final problem size given to the branch-and-bound solver quite small. We use three data reduction rules, which are repeatedly applied to the input data until no more reduction occurs [34]:

1. If $s_i \subseteq s_j$ and $w(s_i) \geq w(s_j)$, then s_i can be removed from consideration. In this case, under any condition for which one might choose s_i, one could choose s_j and be no worse off.
2. If, for all s_i, $e_k \in s_i \Rightarrow e_l \in s_i$, then e_k can be removed from consideration. Any set that covers e_l also covers e_k.

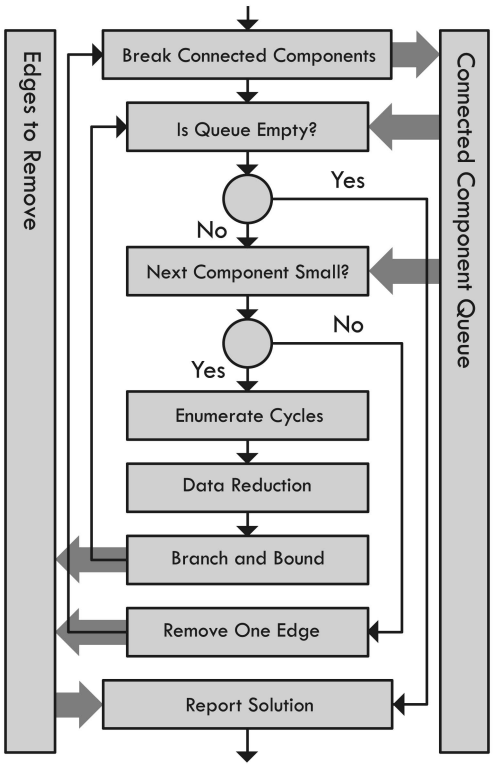

Fig. 4. A schematic of our cycle breaking algorithm. There are two main data elements. $\hat{E}_C$, or the edge set removed to break cycles, is represented by the bar on the left. The queue holding all of the connected components in the graph is represented on the right. The flow of the cycle breaking algorithm is outlined in the center diagram. See the text for details.

3. If element e_k only appears in set s_i, then set s_i must be selected and all elements $e_l \in s_i$ should be removed from consideration as they have been covered by the selection of s_i.

A solution to the problem is an assignment of inclusion or exclusion to each edge. Thus, the search space for the branch-and-bound algorithm is the set of all binary strings of length g, where g is the number of sets in the set cover formulation after data reduction. The branch-and-bound algorithm performs depth-first traversal of the search tree by using the best complete solution thus far to prune subtrees that cannot give a better solution.

8 A Method for Breaking Cycles in a Directed Graph

Definition 8.1. *$\hat{E}_C$ is the feedback arc set resulting from our cycle breaking method. $\hat{G}'$ is the corresponding acyclic subgraph that results.*

Observation 8.2. *Given a directed graph G, the minimum feedback arc set for G is the union of the minimum feedback arc sets for all strongly connected components of G.*

The flow of our cycle breaking algorithm is listed in Fig. 4. First, all strongly connected components of G_A with two or more nodes are put into a queue. $\hat{E}_C$ is initially empty. For each strongly connected component S in the queue, we have two options. If it is small enough, the exact cycle-breaking

method described in Section 7 is used to find the minimum feedback arc set for S, E_C^S. Then, $\hat{E}_C = \hat{E}_C \cup E_C^S$. If S is too large for the exact algorithm to run in a reasonable time, we will use the heuristic algorithm to add a single edge $\hat{e}_{max}$ to $\hat{E}_C$. Removing $\hat{e}_{max}$ might break S into smaller strongly connected components. Therefore, the strongly connected component enumeration algorithm must be run on $S - \hat{e}_{max}$, with the resulting components with two or more nodes enqueued.

Our heuristic algorithm attempts to emulate the $O(\ln d)$ approximation algorithm for set cover. To use the $O(\ln d)$ greedy approximation exactly, we would have to find an edge e_{max} on which the highest number of simple cycles passes through. However, rather than enumerating the potentially exponential number of cycles to discover e_{max}, we will use random cycle walking to look for an edge $\hat{e}_{max}$ that is a likely candidate for e_{max}. To randomly walk a cycle, we first randomly select some starting node s. Then, we perform a randomized depth-first search through the graph until we find a cycle containing s. The path through the DFS tree $\langle s, \ldots, s \rangle$ is a randomly chosen cycle in the graph, although each cycle will not be selected with the same probability. After randomly selecting N cycles and keeping track of the number of these cycles $c(u, v)$ that pass through each edge (u, v), we scan the edges to find an edge with the highest $c(u, v)$. This edge is taken to be $\hat{e}_{max}$ and we remove it from the graph and add it to $\hat{E}_C$.

The algorithm combines the greedy heuristic with the exact solution to find a good solution quickly, invoking the heuristic algorithm only if necessary. If the input maps are reasonably consistent, the heuristic scheme is never used. In this case $\hat{G}' = G'$ and the exact solution is found. However, the heuristic can be invoked when the input contains outliers, that is, markers placed far of order in some map.

9 Software and Experimental Results

We have implemented the proposed method in Java. The software was first run on a suite of synthetic data to establish that the method would be able to find a good consensus in the face of local rearrangements [18]. We have also been able to successfully generate consensus maps for maps generated from seven populations of the crop plant *Zea mays*. We will focus on the maize consensus in this paper.

The software reads the input maps in a tab-delimited text format and outputs the resulting transitively reduced graph in .gdl format, a format specified by the publicly available graph drawing software aiSee, free for academic use (see Fig. 5). The graph encodes information in addition to the consensus order. If no maps disagree on the ordering of an edge, then the edge is solid. If the edge is contested, then the edge is dashed. We display all edges removed during cycle breaking, which we term back edges, in red.

The software creates a separate set of metrics files used for analyzing the result. The metrics files contain metrics computed for each input map and the median. Some useful information that it provides is the number of input, reduction, closure, contested, and back edges. It records the weighted symmetric distance to the median. It outputs which of the nodes in each map act as hubs for contested and back edges and nodes with four or more incident edges of this type are tagged for human scrutiny.

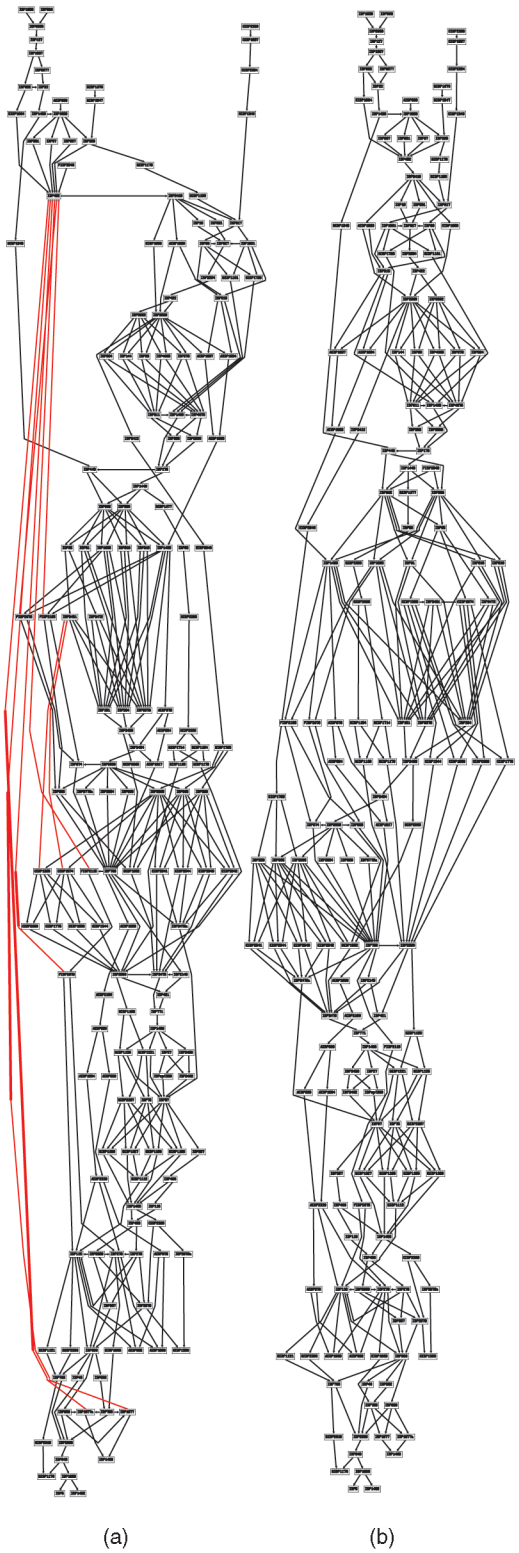

Fig. 5. The consensus map for chromosome 8 as displayed by the graph drawing software aiSee. (a) The median before outliers are removed, with the edges removed during cycle breaking drawn in. (b) The median after removing marker IDP452 from input Map F and IDP2365 from input Map C.

The maize mapping data set used in the study was assembled at the Center for Plant Genomics, Iowa State University, in collaboration with the Laboratory of Abraham

Chromosome	Back	Conflicting	Reduction	Input	Closure	Total	Markers
Chromosome 1	8	15	922	35,303	72,604	78,210	395
Chromosome 2	12	1	662	15,550	34,128	38,781	278
Chromosome 3	0	9	804	17,749	36,792	42,486	291
Chromosome 4	0	0	706	14,013	27,435	31,375	250
Chromosome 5	1	3	840	16,965	38,355	45,150	300
Chromosome 6	19	8	561	11,661	22,939	27,495	234
Chromosome 7	0	4	459	7,531	14,505	18,145	190
Chromosome 8	8	20	362	6,218	13,553	15,931	178
Chromosome 9	0	0	456	7,211	12,911	15,931	178
Chromosome 10	0	1	542	7,980	14,438	18,336	191

Fig. 6. Statistics on the raw maize data, all chromosomes. *Back* edges is a count of the number of edges in E_R, that is, the edges removed to break cycles. *Conflicting* edges report the number of edges with disagreements in order among the inputs. *Reduction* edges are the edges in the transitive reduction of the median graph. The number of edges in the aggregate relation R' are listed as the *input* edges. *Closure* edges are the edges in the transitive closure of the median. The number of edges in a *total* order of *markers* is listed as well. The *total* order edge count can be compared to the *closure* edge count to see how many relations are missing in the consensus. The *closure* edge count can be compared to the *input* edge count to see how much information has been gained in the construction of the median.

Korol, University of Haifa. Six populations of maize were mapped, labeled populations A-F, with each map sharing a large number of markers. The maps also shared markers with the previously mapped IBM population, which we term the IBM map [13], [23]. There have been multiple versions of the consensus maps as the IBM map has been refined over time. An older version of the IBM map was used in conjunction with the population A-F maps to produce the results described in [18].

The software used to create the maps for each line is based on [27]. There are two versions of the genetic map generated by this software. The skeletal map is less complete but more accurate as it consists of the set of markers whose order remains invariant after applying the statistical technique of bootstrapping. The second more complete map adds additional markers by attaching each additional marker to the skeletal marker to which it is closest.

The maize genome has 10 chromosomes and a consensus map for each chromosome was assembled by the software from the complete maps. For chromosomes 3, 4, 7, 9, and 10, the mapping software produced a result without having to do any cycle breaking. In addition, for chromosome 5, only a single edge was removed to create the consensus. However, for chromosomes 1, 2, 6, and 8, the maps showed larger differences resulting in large components being broken by the software, invoking the heuristic algorithm. See Fig. 6 for more details.

By inspecting the resulting graph and looking at the metrics files, we can discover outliers in the data, that is, markers that are responsible for most of the disagreement between a particular graph and the consensus. For example, when inspecting the consensus map for chromosome 2, we see that, of the 12 back edges removed to break cycles, 11 are incident to node IDP618. Additionally, we see from the metric file that map B has the largest distance from the consensus. After inspecting map B, we see that IDP618 is not part of the skeletal map but is one of the markers attached with less certainty to the backbone. Therefore, it seems reasonable that IDP618 might be misordered in map B. After removing this suspect marker from map B, the resulting consensus is cycle free. This underscores the importance of providing information regarding inconsistencies to users. There might be some underlying experimental or methodological reason that one would trust some marker or map over others and one could apply this extra information to tweak the resulting map.

Fig. 7 shows some metrics taken before and after removing outliers from the inputs for five chromosomes. Although the main ordering is unchanged for some chromosomes, we find that, for other chromosomes, the outlier has pulled the ordering of markers away from the most likely consensus location. In this case, the median order produced after the removal of the outlier has a much smaller distance to both the input containing the outlier and the other unedited input maps.

In addition to providing information describing consistency, the software assesses the completeness of the resulting maps. We calculate the number of pairwise relations that would exist in a total order of the markers in the map. The number of edges in the aggregate graph G_A is a count of the number of relations given by the inputs. The number of edges in the transitive closure of the median graph G_M is a count of the number of pairwise relations in the median. We can calculate $new = closure - input$, which is a count of the number of new pairwise relationships generated by the aggregation scheme. We can also calculate $missing = total - closure$, which is a count of the number of missing pairwise relations in the consensus. Fig. 6 shows these numbers on the *Zea mays* data set.

For example, there are 395 mapped markers for chromosome 1 and there would be 78,210 pairwise orderings in the total ordering of these markers. There are only 35,303 known pairwise orderings in the input data, but the consensus contributes an additional 37,301, bringing the total to 72,604. Thus, as expected, finding the consensus results in significant information increase when compared to the set of inputs considered one by one.

We have done some independent verification of the ordering produced by the consensus. This was done by first finding a set of reference markers unordered in the IBM map but mapped in some maps of A-F. These reference markers were then placed into the IBM map by using a wet lab process. We refer to the IBM map containing the reference markers as $\prec_V$.

Chromosome	Back	Conflicting	Reduction	Input	Closure	Bad Markers
Chromosome 1	8	15	922	35303	72604	IDP1964 IDP820
Edited	2	4	946	35193	72261	
Chromosome 2	12	1	662	15550	34128	IDP618
Edited	0	0	680	15497	33692	
Chromosome 3	0	9	804	17749	36792	IDP2435
Edited	0	4	804	17736	36738	
Chromosome 6	19	8	561	11661	22939	IDP1973
Edited	0	2	560	11628	22755	
Chromosome 8	8	20	362	6218	13553	IDP452 IDP2356
Edited	0	3	356	6199	13440	

Fig. 7. In the raw input maps, there were a few outlying markers. These markers were obviously out of order when compared to their positions in the other maps. These markers are identified using a two-step process. First, target markers are identified as those markers that are adjacent to edges labeled as back edges or conflicting edges (described in Fig. 6). Second, the target map is identified as that with the largest weighted symmetric difference distance to the median. This table shows how removing the outlying markers from the target maps affects the consensus. With the targeted removal of these markers, very few disagreements remain among the profile.

Let u be an element in both $\prec_M$ and $\prec_V$. The **comparable set** of elements for some marker u, denoted $V(u)$, is the set $\{v|\neg(u\|_M v) \wedge \neg(u\|_V v)\}$. The **agreeing set** of elements $A(u)$ is the set $\{v|v \in V \wedge u \prec_M v \Leftrightarrow u \prec_V v\}$. In total, 96 markers were used for verification. Of the 96, 85 showed perfect agreement when compared to the consensus $V(u) = A(u)$. As shown in Fig. 8, only two markers showed less than 95 percent of agreement. When taken as a whole, the reference markers showed over 99.5 percent of agreement.

10 Discussion

We note that finding a median relation under the weighted symmetric difference distance is NP-hard, but, for small connected components, the problem can be exactly solvable. For this reason, we implement an exact solver that works as long as the connected components to be solved in the cycle breaking problem are small. Additionally, we implement a heuristic solver that is invoked when this is not the case. We heuristically solve the problem edge by edge only until that point at which the exact solver can again be employed.

Name	$\|V(u)\|$	$\|A(u)\|$	Percent
AIDP935	234	233	99.57
CIDP1265	221	219	99.10
BIDP1538	149	147	98.66
AIDP2162	99	97	97.98
AIDP2167	99	97	97.98
AIDP908	99	97	97.98
BIDP1048	71	68	95.77
CIDP2536	127	110	86.61
CIDP2036	138	110	79.71

Fig. 8. Of the 96 markers occurring both in the consensus map and the verification map, 85 showed a perfect placement. Listed are those that show disagreement between the two maps. For each marker u, the table lists the number of markers v for which the ordering of u and v can be determined in both maps. It also lists the number of markers for which the ordering agrees, as well as the percent agreement. Of the 96 markers verified, only two show less than 95 percent of agreement.

We wish the viewer of the map to be informed of the decisions that we make in coming to the consensus and, for this reason, we calculate the information in addition to the order itself. We show how we use this information to find outliers in the data. The outcome represents an average of the inputs and can be affected by such outliers. For this reason, we remove the outliers and run the method the second time, producing the final result. These outliers are also presented with the result so that the data generation team can analyze why they might have been placed out of order.

Areas of the consensus map that have significant ambiguity can be identified by analyzing paths between node pairs. If there are many paths or multiple long independent paths between a pair of nodes, then there is a lot of missing information about the relative ordering of nodes on these different paths. Therefore, we can use the consensus to prioritize laboratory experiments that would fill in missing information such that the most ambiguous areas of the map are fixed first.

We use one of the many weighting schemes that one could envision by basing the edge weight on the path length between two markers. We provide proofs that give sufficient conditions for the median to satisfy various criteria. Any weight function that satisfies the prescribed conditions would result in a median that satisfies these criteria. One could, for example, base the edge weight on the genetic distance between markers. One could weight each map based on the perceived quality (for example, the number of individuals in the mapping population) and use this as a multiplicative factor in the edge weights of each map. The goal of these weighting schemes would be to produce a higher quality consensus.

Some uses of a genetic map need distance information in addition to the ordering information that we describe here. Applying edge distances to the consensus is not a straightforward task as distances between two genetic maps are not directly comparable [39]. For this reason, some sort of complex normalization would be needed before even considering using distances within the context of creating consensus maps. This is an area in need of further research.

11 Conclusions

We have formulated consensus genetic map creation as finding a median partial order using the weighted symmetric difference distance. We have shown that a median found using this metric satisfies the extended Condorcet, unanimity, and positive responsiveness criteria, that is, criteria also satisfied by the Kemeny median, which is widely used in the field of social choice. Finding the median by using this distance is NP-hard.

We model each input map as a partial order and then assign weights to each relationship $u \prec v$ based on the shortest path distance between u and v in the transitively reduced graph corresponding to the order. We chose our weight function with the idea that we are more certain of the order of distant markers than close markers in genetic maps.

We model the cycle-breaking problem that arises in finding a median as the set cover problem. Despite this problem being NP-hard, we have been able to solve this exactly by using data reduction in many cases. In the case where the problem is not exactly solvable, we use a heuristic algorithm based on the $O(\ln d)$ approximation of the set cover, where d is the size of the largest set.

We have validated our implementation by constructing consensus maps for seven populations of the crop plant maize. The correctness of the ordering produced by these maps has been verified independently in the wet lab, with a 99.5 percent agreement rate between the consensus map and the validation map on the placement of the reference markers.

Acknowledgments

The authors thank Tsui-Jung Wen, Hsin Debbie Chen, Olga Alechina, Olga Ellern, Elizabeth Hahn, Efim Ronin, Josh Shendelman, Diane Sickau, and Natalija Zazubovits for their assistance in generating mapping data for maize. They also thank David Fernandez-Baca for the discussion that first led to investigating the relevance of social choice literature, Pang Ko, Scott Emrich, and Dan Ashlock for their input and feedback, and the referees for their many helpful comments. This project was supported by the US National Science Foundation under Grants DBI-9975868, DBI-0321711, and DBI-0527192, the Binational Agricultural Research and Development Program (Project US-3873-06), the Hatch Act, and State of Iowa funds.

References

[1] A.V. Aho, M.R. Garey, and J.D. Ulman, "The Transitive Reduction of a Directed Graph," *SIAM J. Computing,* vol. 1, pp. 131-137, 1972.
[2] N. Ailon, M. Charikar, and A. Newman, "Aggregating Inconsistent Information: Ranking and Clustering," *Proc. 37th Ann. ACM Symp. Theory of Computing,* pp. 684-693, 2005.
[3] K. Arrow, *Social Choice and Individual Values.* John Wiley, 1951.
[4] J.P. Barthelemy and B. Monjardet, "The Median Procedure in Data Analysis: New Results and Open Problems," *Proc. Second Conf. Int'l Federation of Classification Societies,* pp. 309-316, 1988.
[5] T.H. Corman, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms,* second ed. 2003.
[6] A. Davenport and J. Kalagnanam, "A Computational Study of the Kemeny Rule for Preference Aggregation," *Proc. Nat'l Conf. Artificial Intelligence,* 2004.
[7] C. Demetrescu and F.I. Giuseppe, "Trade-Offs for Fully Dynamic Transitive Closure on DAGs: Breaking through the $O(n^2)$ Barrier," *J. ACM,* vol. 52, pp. 147-156, 2005.
[8] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank Aggregation Methods for the Web," *Proc. 10th Int'l Conf. World Wide Web,* 2001.
[9] P. Eades, X. Lin, and W.F. Smith, "A Fast and Effective Solution for the Feedback Arc Set Problem," *Information Processing Letters,* vol. 47, pp. 319-323, 1998.
[10] G. Even, J. Naor, B. Schieber, and M. Sudan, "Approximating Minimum Feedback Sets and Multicuts in Directed Graphs," *Algorithmica,* vol. 20, pp. 151-174, 1998.
[11] C.T. Falk, "Preliminary Ordering of Multiple Linked Loci Using Pairwise Linkage Data," *J. Quantum Trait Loci,* vol. 2, 1992.
[12] R.W. Floyd, "Algorithm 97: Shortest Path," *Comm. ACM,* vol. 5, p. 345, 1962.
[13] Y. Fu, T.J. Wen, Y.I. Ronin, H.D. Chen, L. Guo, D.I. Mester, Y. Yang, M. Lee, A.B. Korol, D.A. Ashlock, and P.S. Schnable, "Genetic Dissection of Intermated Recombinant Inbred Lines Using a New Genetic Map of Maize," *Genetics,* vol. 174, pp. 1671-1683, 2006.
[14] A. Goralcikova and K. Koubek, "A Reduct-and-Closure Algorithm for Graphs," *Math. Foundations of Computer Science,* vol. 74, pp. 301-307, 1979.
[15] S. Hannenhalli and P.A. Pevzner, "Transforming Men into Mice (Polynomial Algorithm for Genomic Distance Problem)," *Proc. 36th Ann. IEEE Symp. Foundations of Computer Science,* pp. 581-592, 1995.
[16] S. Hannenhalli and P.A. Pevzner, "Transforming Cabbage into Turnip (Polynomial Algorithm for Sorting Signed Permutations by Reversals," *J. ACM,* vol. 48, pp. 1-27, 1999.
[17] O. Hudrey, "Computation of Median Orders: Complexity Results," *Proc. DIMACS-LAMSADE Workshop Computer Science and Decision Theory,* 2004.
[18] B. Jackson, S. Aluru, and P. Schnable, "Consensus Genetic Maps: A Graph Theoretic Approach," *Proc. IEEE Computational Systems Bioinformatics Conf.,* pp. 35-43, 2005.
[19] D.B. Johnson, "Finding All the Elementary Circuits of a Directed Graph," *SIAM J. Computing,* vol. 4, pp. 77-84, 1975.
[20] J.P. Kemeny, "Mathematics without Numbers," *Daedelus,* vol. 88, pp. 577-591, 1959.
[21] S. Knapp, C. Echt, and B.H. Liu, "Genome Mapping with Non-Inbred Crosses Using Gmendel 2.0," *Maize Genetics Cooperation Newsletter,* vol. 66, pp. 22-79, 1992.
[22] E. Lander, P. Green, J. Abrahamson, A. Barlow, M.J. Daly, S.E. Lincoln, and L. Newburg, "An Interactive Computer Package for Constructing Primary Genetic Linkage Maps of Experimental and Natural Populations," *Genomics,* vol. 1, pp. 174-181, 1997.
[23] M. Lee, N. Sharopova, W.D. Beavis, D. Grant, M. Katt, D. Blair, and A. Hallauer, "Expanding the Genetic Map of Maize with Intermated B73 Mo17 (IBM) Population," *Plant Molecular Biology,* vol. 48, pp. 453-461, 2002.
[24] A. Levenglick, "Fair and Reasonable Election Systems," *Behavioral Science,* vol. 20, pp. 34-46, 1975.
[25] L. Lovasz, "On the Ratio of Optimal Integral and Fractional Covers," *Discrete Math.,* vol. 13, pp. 383-390, 1964.
[26] D. Mester and O. Braysy, "Fast and High Precision Algorithms for Optimization in Large-Scale Genomic Problems," *Computational Biology and Chemistry,* vol. 28, pp. 281-289, 2004.
[27] D. Mester, E. Ronin, E. Nevo, and A. Korol, "Constructing Large Scale Genetic Maps Using Evolutionary Strategy Algorithm," *Genetics,* vol. 165, pp. 2269-2282, 2003.
[28] M.E. Moret, J. Tang, and T. Warnow, "Reconstructing Phylogenies from Gene-Content and Gene-Order Data," *Math. Evolution and Phylogeny,* O. Gascuel, ed., chapter 12, pp. 321-352, Oxford Univ. Press, 2006.
[29] J.M. Olson and M. Boehnke, "Monte Carlo Comparison of Preliminary Methods of Ordering Multiple Genetic Loci," *Am. J. Human Genetics,* vol. 47, pp. 470-482, 1990.
[30] J. Ott, *Analysis of Human Genetic Linkage.* John Hopkins Univ. Press, 1985.
[31] D. Sankoff, C. Zheng, and A. Lenert, "Reversals of Fortune," *Proc. RECOMB Workshop Comparative Genomics,* pp. 131-141, 2005.
[32] T. Schiex and C. Gaspin, "CARTHAGENE: Constructing and Joining Maximum Likelihood Genetic Maps," *Proc. Fifth Int'l Conf. Intelligent Systems in Molecular Biology,* vol. 48, pp. 453-461, 1997.
[33] P. Slavik, "A Tight Analysis of the Greedy Algorithm for Set Cover," *Proc. 28th Ann. ACM Symp. Theory of Computing,* pp. 435-441, 1996.

[34] M. Syslo, N. Deo, and J. Kowalik, *Discrete Optimization Algorithms and Pascal Programs.* Prentice Hall, 1983.
[35] M. Truchon, "An Extension of the Condorcet Criterion and Kemeny Orders," cahier 98-15 du Centre de Recherche en Economie et Finance Appliquees, 1998.
[36] M. Truchon, "Aggregation of Rankings in Figure Skating," *Cahiers de Rcherche,* 2005.
[37] Y. Wakabayashi, "The Complexity of Computing Medians of Relations," *Resenhas,* vol. 3, pp. 323-349, 1998.
[38] S. Yancopoulos, O. Attie, and R. Friedberg, "Efficient Sorting of Genomic Permutations by Translocation, Inversion, and Block Interchange," *Bioinformatics,* vol. 21, pp. 3340-3346, 2005.
[39] H. Yao and P.S. Schnable, "Cis-Effects on Meiotic Recombination Across Distinct a1-sh2 Intervals in a Common Zea Genetic Background," *Genetics,* vol. 170, pp. 1929-1944, 2004.
[40] I.V. Yap, D. Schneider, J. Kleinberg, D. Matthews, S. Cartinhour, and S.R. McCough, "A Graph-Theoretic Approach to Comparing and Integrating Genetic Physical and Sequence-Based Maps," *Genetics,* vol. 165, pp. 2235-2247, 2003.

Benjamin N. Jackson received the BS degree in computer science from Iowa State University (ISU) in 2001. He is currently working toward the PhD degree in the Department of Electrical and Computer Engineering at ISU, where he is a recipient of a USDA-MGET Interdisciplinary Fellowship. Prior to studying at ISU, he was a developer for the IBM Websphere Product in Rochester, Minnesota. His research interests include computational biology, parallel algorithms, and scientific computing.

Patrick S. Schnable received the BS degree from Cornell University in 1981 and the PhD degree from Iowa State University in 1986 for his studies (with Peter Peterson) of the Mu transposon. Following a postdoctoral appointment with Heinz Saedler at the Max Planck Institute, Koeln, Germany, he was appointed in 1988 to the faculty at Iowa State University, where he is currently a professor in the Department of Agronomy and Genetics and the Department of Development and Cell Biology, the associate director of the Plant Sciences Institute, and the founding director of the Center for Plant Genomics. He serves on a variety of scientific advisory boards and is an elected member of the American Association for the Advancement of Science Section Committee of the Agriculture, Food and Renewable Resources Section and is the chair of the Maize Genetics Executive Committee. He is an active participant in interdisciplinary graduate training programs, including the Interdepartmental Genetics, Bioinformatics and Computational Biology, Interdepartmental Plant Physiology and the Molecular, Cellular and Developmental Biology Graduate Programs. He manages a vigorous research program that emphasizes interdisciplinary approaches to understanding plant biology. His own expertise is on genetics, molecular biology, and genomics, but he collaborates with researchers in diverse fields, including biochemistry, plant breeding, plant physiology, bioinformatics, computer science, and engineering. His research interests include the maize genome structure, heterosis, meiotic recombination, cytoplasmic male sterility, cuticular wax biosynthesis, and the development of new genomic technologies and bioinformatics approaches.

Srinivas Aluru received the BTech degree in computer science from the Indian Institute of Technology, Chennai, India, in 1989 and the MS and PhD degrees in computer science from Iowa State University in 1991 and 1994, respectively. He is the Stanley Chair of Interdisciplinary Engineering and a professor of electrical and computer engineering at Iowa State University. He is a member of the Center for Plant Genomics, LH Baker Center for Bioinformatics and Biological Statistics, and Bioinformatics and Computational Biology Graduate Program, which he formerly chaired. Earlier, he held faculty positions at New Mexico State University and Syracuse University. From 2004 to 2006, he was an IEEE Computer Society Distinguished Visitor. He has served on numerous program committees and has served in leadership roles for several conference and workshops in parallel processing and computational biology, including serving as the program chair of the HiPC '07 and the program vice chair for IPDPS '07, ICPP '07, HiPC '06, and SC '03. He is a cochair of HiCOMB (http://www.hicomb.org) and a coeditor of several special issues on this topic. He is the editor of a comprehensive handbook on computational molecular biology published in 2005. His research interests include parallel algorithms and applications, bioinformatics and computational biology, and combinatorial scientific computing. His contributions to computational biology are computational genomics, string algorithms, and parallel methods for solving large-scale problems arising in biology. He is a member of the ACM, SIAM, ISCB, and Life Sciences Society and a senior member of the IEEE and Ithe EEE Computer Society. He received a US National Science Foundation CAREER Award in 1997, the IBM Faculty Award in 2002, the Iowa State University Foundation Award for his midcareer achievement in research in 2006, the Warren B. Boast Undergraduate Teaching Award in 2005, and the Best Paper Awards from IPDPS '06 and CSB '05.