Efficient and Accurate Construction of Genetic Linkage Maps from Noisy and Missing Genotyping Data

Yonghui Wu[1], Prasanna Bhat[2], Timothy J. Close[2], and Stefano Lonardi[1,⋆]

[1] Dept. of Computer Science and Eng., University of California, Riverside, CA
[2] Dept. of Botany & Plant Sciences, University of California, Riverside, CA
stelo@cs.ucr.edu

Abstract. We introduce a novel algorithm to cluster and order markers on a genetic linkage map, which is based on several theoretical observations. In most cases, the true order of the markers in a linkage group can be efficiently computed from the minimum spanning tree of a graph. Our empirical studies confirm our theoretical observations, and show that our algorithm consistently outperforms the best available tool in the literature, in particular when the genotyping data is noisy or in case of missing observations.

1 Introduction

Genetic linkage mapping dates back to the early 20th century when scientists began to understand the recombinational nature and cellular behavior of chromosomes. In his landmark paper published in 1913, Sturtevant studied the first genetic linkage map of chromosome X of *Drosophila melanogaster* [19]. Ever since its introduction, genetic linkage mapping has been a cornerstone of a variety of applications in biology, including map-assisted breeding, disease association analysis and map-assisted gene cloning, just to name a few.

Genetic linkage maps historically began with just a few to several tens of phenotypic markers obtained one by one by observing morphological and biochemical variations of an organism, mainly following mutation. During the past few decades the introduction of DNA-based markers such as RFLPs, RAPDs, SSRs and AFLPs caused genetic maps to become much more densely populated, generally into the range of several hundred to more than 1,000 markers per linkage map. Most recently, the accumulation of sequence information has led to a further leap in marker density, principally driven by very high throughput and highly accurate genotyping that can accommodate thousands, or even hundreds of thousands, of simultaneous genotyping reactions by one person in a single day. In plants, one of the most densely populated maps is that of *Brassica napus* [20], which consists of 13,551 markers. High density genetic maps do not require complete genome sequencing but rather are a critical step in the study of organisms for which the whole genome sequence is unlikely to be available in the near future.

A genetic map is a linear ordering of markers (also called *loci*) along the chromosome. The map is built using input data typically composed of the states of the loci on a set of individuals obtained from controlled crosses. When an order of the markers is computed from the data, the genetic distance between nearby markers can be relatively

[⋆] Corresponding author. This project was supported in part by NSF CAREER IIS-0447773 and NSF DBI-0321756.

R. Giancarlo and S. Hannenhalli (Eds.): WABI 2007, LNBI 4645, pp. 395–406, 2007.

easily estimated. In order to characterize the quality of an order, various objective functions have been proposed in the literature, e.g., *minimum Sum of Square Errors* [18], *minimum number of recombination events* [16], *Maximum Likelihood* [11], *maximum Sum of adjacent LOD scores* [21], *minimum Sum of Adjacent Recombination Fractions* [3], *minimum Product of Adjacent Recombination Fractions* [22].

Searching for an optimal order with respect to any of the objective functions mentioned above is computationally difficult. Enumerating all the possible orders quickly becomes infeasible since the total number of distinct orders is proportional to $n!$, which is too large even if n is rather small. With the exception of SSE, the rest of the objective functions listed above can be decomposed into a simple sum of terms involving only pairs of markers. Liu [14] first observed the connection between the marker ordering problem and the traveling salesman problem. Various searching heuristics that were originally developed for the TSP problem, such as *simulated annealing* [12], *genetic algorithms* [8], *tabu search* [6,7], *ant colony optimization*, and iterative heuristics such as *K-opt* and *Lin-Kernighan heuristic* [13] have been applied to the genetic mapping problem in various computational packages. For example, JOINMAP [11] implements simulated annealing, CARTHAGENE [17,2] uses a combination of Lin-Kernighan heuristic, tabu search and genetic algorithms, ANTMAP [10] exploits the ant colony optimization heuristic, [5] is based on genetic algorithms, and [15] takes advantage of evolutionary algorithms. Finally, RECORD [16] implements a combination of greedy and Lin-Kernighan heuristic.

Most of the algorithms proposed in the literature for genetic linkage mapping find reasonably good solutions. Nonetheless, they fail to identify and exploit the combinatorial structures hidden in the data. Some of them simply start to explore the space of the solutions from a purely random order (see, e.g., [17,15,11,10]), while other start from a simple greedy solution (see, e.g., [16,18]). In this paper, we will show both theoretically and empirically, that when the data quality is high, the optimal order can be identified via the simple minimal spanning tree algorithm. We will also show that when the data is noisy or data is missing, our algorithm consistently constructs better genetic maps than the best available tools in the literature.

2 Basic Concepts and Notations

First we introduce some basic concepts of genetics and establish a common notation. A *single nucleotide polymorphism* (SNP) is a variation on a chromosome where a single nucleotide (A, C, G, T) differs between members of a species or between paired chromosomes in an individual. SNP sites can be used as *genetic markers*.

The organisms considered here are an ideal case of fully homozygous (or nearly so) diploids derived from two highly inbred (fully homozygous) parents. In this system, for every locus there is one maternal and one paternal allele and with rare exception each locus exists in only two possible fully homozygous states distinguished by two alternative nucleotides. By convention, the two states are denoted as `A` or `B`. A genetic marker is said to be *homozygous* if the two alleles at a given locus have the same state, and it is said to be *heterozygous* otherwise.

Various population types have been studied in association with the genetic mapping problem, which includes *BackCross* (BC1), *Doubled Haploid* (DH), *Haploid* (Hap), *Recombinant Inbred Line* (RIL), etc. Our algorithm can handle Hap, advanced RIL (low heterozygosity) and BC1 populations in addition to DH's. In what follows, we will concentrate on the DH population, but the extension to the Hap, advanced RIL and BC1 populations is straightforward.

Briefly, a DH population for genetic map construction is prepared as follows. Let M be the set of markers of interest. Pick two highly inbred (fully homozygous) parents p_1 and p_2. We assume that the parents p_1 and p_2 are homozygous on every markers in M (those markers that are heterozygous in either p_1 or p_2 are simply excluded from consideration), and the same marker always has different allelic states in the two parents (those markers having the same allelic state in both parents are also excluded from M). By convention, we use symbol `A` to denote the allelic states appearing in p_1 and `B` to denote the allelic states appearing in p_2. Parent p_1 is crossed with parent p_2 to produce the first generation, called *F1*. The individuals in the F1 generation are heterozygous for every marker in M, with one chromosome being all `A` and the other chromosome being all `B`.

In the DH system, gametes produced by meiosis from the F1 generation are fixed in a homozygous state by doubling the haploid chromosomes to produce a doubled haploid individual (hence the name doubled haploid). The doubled haploid individuals, denoted by N, are then genotyped on the set M of markers, i.e., the state of each marker is determined in a wet lab experiment. The genotypes are either homozygous `A` or homozygous `B`.

The genotyping data which will be fed into our algorithm is collected into a matrix $\mathbb{A}$ of size $m \times n$, where $m = |M|$ and $n = |N|$. Each row of $\mathbb{A}$ corresponds to a marker in M, and each column of $\mathbb{A}$ corresponds to an individual in N. Given a marker $l_i \in M$, we use $\mathbb{A}[i,]$ to refer to the row corresponding to l_i. Given an individual $c_k \in N$, we use $\mathbb{A}[,k]$ to refer to the column corresponding to c_k. Each entry in the matrix can be either `A` (i.e., the same allelic state of its parent p_1) or `B` (i.e., the same allelic state of its parent p_2). For the time being, we assume there is no missing observation in the matrix. The case where there is missing data will be discussed later in the paper.

Building a genetic map from the matrix $\mathbb{A}$ is a two-step process. First, one has to partition the markers in $\mathbb{A}$ into groups, each of which corresponds to a chromosome. More specifically, one needs to determine which markers are from the same *linkage group*[1]. This problem is essentially a clustering problem. Second, given a set of markers in the same linkage group, one needs to determine their correct order on the chromosome.

For a pair of markers $l_1, l_2 \in M$ and an individual $c \in N$, we say that c is a *recombinant* with respect to l_1 and l_2 if c has genotype `A` on l_1 and genotype `B` on l_2 (or vice versa). If l_1 and l_2 are in the same linkage group, then a recombinant is produced if an odd number of crossovers occurred between the paternal chromosome and the maternal chromosome within the region spanned by l_1 and l_2 during meiosis. We denote with $\mathbf{P}_{i,j}$ the probability of a recombinant event with respect to a pair of markers (l_i, l_j). $\mathbf{P}_{i,j}$ varies from 0.0 to 0.5 depending on the distance between l_i and l_j. At one

[1] A *linkage group* is group of loci known to be physically connected, that is, they tend to act as a single group (except for recombination of alleles by crossing-over) in meiosis instead of undergoing independent assortment. Ideally, each linkage group corresponds to one chromosome, but sometimes multiple LGs can reside on the same chromosome if they are too far apart.

extreme, if l_i and l_j belong to different LGs, then $\mathbf{P}_{i,j} = 0.5$ because alleles at l_i and l_j are passed down to next generation independently from each other. At the other extreme, when the two markers l_i and l_j are so close to each other that no recombination can occur between them, then $\mathbf{P}_{i,j} = 0.0$.

Let (l_i, l_j) and (l_p, l_q) be two pairs of markers from the same linkage group. We say that (l_i, l_j) is *enclosed* in (l_p, l_q) if the region of the chromosome spanned by l_i and l_j is fully contained in the region spanned by l_p and l_q. A fundamental law in Genetics is that if (l_i, l_j) is fully contained in (l_p, l_q) then $\mathbf{P}_{i,j} \leq \mathbf{P}_{p,q}$.

The total number of recombinants in N with respect to the pair (l_i, l_j) can be easily determined by computing the Hamming distance $d_{i,j}$ between row $\mathbb{A}[i,]$ and row $\mathbb{A}[j,]$. It is easy to prove that $d_{i,j}/n$ the *maximum likelihood estimate* (MLE) for $\mathbf{P}_{i,j}$.

3 Some Theoretical Observations

Our first observation is that when two markers l_i and l_j belong to two different linkage groups, then $d_{i,j}$ will be large with high probability. This is formally captured in the following theorem. Recall that in this case, $\mathbf{P}_{i,j} = 0.5$.

Theorem 1. *Let l_i and l_j be two markers that belong to two different LGs, and let $d_{i,j}$ be the Hamming distance between $\mathbb{A}[i,]$ and $\mathbb{A}[j,]$. Then,*

$$E(d_{i,j}) = n/2 \quad \textit{and} \quad \mathbf{P}(d_{i,j} < \delta) \leq e^{\frac{2(n/2-\delta)^2}{n}}$$

where $\delta < n/2$.

Proof. Let $c_k \in N$ and let $X^k_{i,j}$ be a random indicator variable which is equal to 1 if c_k is a recombinant with respect to l_i and l_j and to 0 otherwise. Clearly $E(X^k_{i,j}) = \frac{1}{2}$, and $d_{i,j} = \sum_k X^k_{i,j}$. The family of random variables $\{X^k_{i,j} : 1 \leq k \leq n\}$ are i.i.d. According to linearity of expectation, $E(d_{i,j}) = n/2$. The bound $\mathbf{P}(d_{i,j} < \delta) \leq e^{\frac{2(n/2-\delta)^2}{n}}$ derives directly from Hoeffding's inequality [9]. □

Theorem 1 allows us to partition the markers into linkage groups fairly easily. The algorithmic details will be presented in the next section. In the following, we will assume that the markers have been successfully clustered into linkage groups, and we will focus on the ordering problem only.

Let us assume now that all the markers in M belong to the same linkage group. Let $G(M, E)$ be an edge-weighted complete graph on the set of vertices M. The weight of an edge $(l_i, l_j) \in E$ is set to $\mathbf{P}_{i,j}$, which is the pairwise recombinant probability between the corresponding markers. A *traveling salesman path* (TSP path) $\mathit{\Gamma}$ in G is a path that visits every marker/vertex once and only once[2]. The weight $w(\mathit{\Gamma})$ of a TSP path $\mathit{\Gamma}$, is simply the sum of the weights of the edges on $\mathit{\Gamma}$. Since each TSP path $\mathit{\Gamma}$ defines an order $\mathit{\Pi}$ of the markers (and vice versa), we define the *weight* of an order $\mathit{\Pi}$ as the weight of the corresponding TSP path $\mathit{\Gamma}$ in G. A linear ordering of the markers is also called a *map* of the markers.

[2] Note the difference between a traveling salesman path and a traveling salesman tour. A tour is a cycle (i.e., the salesman returns back to the origin).

Lemma 1. *Let Π_0 be the true order of markers (according to their positions on the chromosome). Then, the weight of Π_0 is minimum among all the possible orders of M.*

Proof. Let $l_1, l_2, \ldots, l_m$ be the markers in M in true order Π_0, and let $\Pi_1 = l_{i_1}, l_{i_2}, \ldots, l_{i_m}$ be any order. We have $w(\Pi_0) = \sum_{2 \leq i \leq m} \mathbf{P}_{i-1,i}$ and $w(\Pi_1) = \sum_{2 \leq j \leq m} \mathbf{P}_{i_{j-1},i_j}$. Let $S_0 = \{\mathbf{P}_{i-1,i} | 2 \leq i \leq m\}$ and $S_1 = \{\mathbf{P}_{i_{j-1},i_j} | 2 \leq j \leq m\}$. Now observe that there is a one to one correspondence between the elements in S_0 and the elements in S_1, such that if $\mathbf{P}_{i-1,i} \in S_0$ is mapped to $\mathbf{P}_{i_{j-1},i_j} \in S_1$ then the pair of markers (l_{i-1}, l_i) is fully contained in the pair $(l_{i_{j-1}}, l_{i_j})$, and hence $\mathbf{P}_{i-1,i} \leq \mathbf{P}_{i_{j-1},i_j}$. Therefore, we conclude that $w(\Pi_0) \leq w(\Pi_1)$. □

According to Lemma 1, in order to determine the correct order of the markers, one has to find the minimum weight TSP path in G. Although the problem of finding the minimum weight TSP path in a general graph is NP-complete [4], in our case it is rather easy, as shown next. Recall that a *minimum (weight) spanning tree* (MST) of G is a subgraph of G which is a tree that spans all the vertices of G and has minimum total weight. To be technically accurate, we assume that the graph G has only one minimum weight spanning tree.

Lemma 2. *Let Γ_0 be the MST of G. Then, Γ_0 is also the minimum weight TSP path.*

Proof. Let $l_1, l_2, \ldots, l_m$ be the markers in their true order. Let us run Prim's minimum spanning tree algorithm [1] on G starting from the first marker in the linkage group, i.e., l_1. Prim's algorithm iteratively adds node (and edges) to a partially discovered tree until all the nodes are included. The next node to be added is the closest one to the partially discovered tree. Let l_{i-1} be the node added in the previous step of Prim. Due to the way the edge weights are assigned in G, the next marker to be added will be l_i. Therefore, the MST is also a TSP path in G. □

Theorem 2. *The true order of the markers in M can be determined by computing the MST in G.*

Proof. Follows directly from Lemma 1 and 2. □

Theorem 2 claims that the true order of the markers in a linkage group can be identified by simply running Prim's MST algorithm (or any other MST algorithms) on the graph G, which would take quadratic time in the number of markers. Unfortunately, we do not known the exact pairwise recombinant probabilities $\mathbf{P}_{i,j}$. What we have are their maximum likelihood estimates $d_{i,j}/n$ for those probabilities. Thus, we replace $\mathbf{P}_{i,j}$ by $d_{i,j}$ as the edge weights in G, and we call H the resulting graph.

Our objective is to find a minimum weight TSP path in the graph H (which turns out to be the same objective function as used in [16]). When $n \to \infty$, the max likelihood estimates converge to the true probabilities $\mathbf{P}_{i,j}$. According to Lemma 1, the minimum weight TSP path will reveal the true order of the markers. Thus, we run Prim's algorithm on H to compute the optimum spanning tree. If the MLEs are accurate, according to Lemma 2, the MST will be a TSP path. In practice, due to noise in the genotyping data or due to an insufficient number of individuals, the spanning tree may not be a path – but hopefully "very close" to a path. As we will show in Section 5, this is exactly what we observed when running our algorithm on both real data and simulated data – the MST produced is "almost" a path. In any case, when a tree is not yet a path, we will try to transform it into a path, as explained next.

4 Algorithmic Methods

The construction of a genetic linkage map consists of two steps. In the first step, one clusters the markers into linkage groups. In the second, one determines the order of the markers in each LG.

4.1 Clustering Markers into Linkage Groups

In order to cluster the markers into linkage groups, we construct a complete graph $H(M, E)$ over the set of markers to be clustered. The weight of an edge $(l_i, l_j) \in E$ is equal to the pairwise distance $d_{i,j}$ between l_i and l_j. As shown in Theorem 1, if two markers belong to different LGs, then the distance between them will be large with high probability. Once a small probability ϵ is chosen by the user (the default is ϵ=0.000001), we can determine δ by solving the equation $2(n/2 - \delta)^2/n = \log_e \epsilon$. We then remove all the edges from $H(M, E)$ whose weight is larger than or equal to δ. The resulting graph will break up into connected components, each of which is assigned to a linkage group.

The parameter ϵ should be chosen according the quality of the data. In practice, this is not such a critical issue since the recombinant probability between nearby markers on the same linkage group is usually very small (less than 0.05). According to our experience, our algorithm is capable of determining the correct number of LGs for a fairly large range of values for ϵ.

4.2 Ordering Markers in Each Linkage Groups

Before ordering the markers in each linkage groups we preprocess the data by collecting cosegregating markers[3] into *bins*. Each bin is uniquely identified by one of its members. Given one of the linkage groups obtained by the step above, we first construct a complete graph $H(B, E)$, where B corresponds to the bins in that linkage group, and the weight on the edges is the pairwise distance between the corresponding representative markers.

As mentioned earlier, in order to find a good TSP path in $H(B, E)$, we start by constructing an MST. If the MST turns out to be a path, we are done. Otherwise, we need to transform the MST into a path, in a way so that the ordering captured by the tree is maintained as much as possible. We proceed as follows. First, we find the longest path in the MST, hereafter referred to as the *backbone*. The bins/vertices that do not belong to the path will be first disconnected from it. Then, the disconnected bins/vertices will be re-inserted into the backbone one by one at the position where the resulting backbone has the minimum total weight. The path resulting at the end of this process is our initial solution, which might not be locally optimal.

Once the initial solution is computed, we apply two heuristics that iteratively perform local perturbations to it in an attempt to improve its quality. First, we use the commonly-used K-opt ($K = 2$ in this case) heuristic. We cut the current path into three pieces, and try all the possible rearrangements of the three pieces. If any of the resulting paths has a less total weight, it will be saved. This procedure is repeated until no further improvement is possible. In the second heuristic, we try to relocate each node in the

[3] *Cosegregating* markers are those markers for which the pairwise distances is 0.

path to all the other possible positions. If this relocation reduces the weight, the new path will be saved. We apply the 2-opt heuristic and the relocation heuristic iteratively until none of them can further reduce the weight of the path. The resulting TSP path represents our final solution.

4.3 Dealing with Missing Data

In our discussion so far, we assumed no missing data. This assumption is, however, not very realistic. As it turns out in practice, it is rather common to have missing data about the state of a marker. In fact, as we will show in our experimental results, missing observations do not have as much negative impact on the accuracy of the final map as genotype errors. Thus, it appears beneficial to leave uncertain genotypes as missing observations than arbitrarily calling them one way or the other.

We deal with missing observations via an *Expectation Maximization* (EM) algorithm. Observe that if we knew the order of the markers (or, bins, if we have cosegregating markers), the process of imputing the missing data would be relatively straightforward. For example, suppose we knew that marker l_3 immediately follows marker l_2, and that l_2 immediately follows marker l_1. Let us denote with $\hat{\mathbf{P}}_{1,2}$ the estimate of the recombinant probabilities between markers l_1 and l_2, and with $\hat{\mathbf{P}}_{2,3}$ the recombinant probability between markers l_2 and l_3. Let us assume that for an individual c the genotype at locus l_2 is missing, but the genotypes at loci l_1 and l_3 are available. Without loss of generality, let us suppose that they are both A. Then, the posterior probability for the genotype at locus l_2 in individual c is

$$\mathbf{P}\{\text{genotype in } c \text{ at } l_2 \text{ is } \mathtt{A}\} = \frac{(1-\hat{\mathbf{P}}_{1,2})(1-\hat{\mathbf{P}}_{2,3})}{(1-\hat{\mathbf{P}}_{1,2})(1-\hat{\mathbf{P}}_{2,3})+\hat{\mathbf{P}}_{1,2}\hat{\mathbf{P}}_{2,3}}$$

and $\mathbf{P}\{\text{genotype in } c \text{ at } l_2 \text{ is } \mathtt{B}\} = 1 - \mathbf{P}\{\text{genotype in } c \text{ at } l_2 \text{ is } \mathtt{A}\}$. This posterior probability is the best estimate for the genotype of the missing observation. Similarly, one can compute the posterior probabilities for different combinations of the genotypes at loci l_1 and l_2.

In order to deal with uncertainties in the data, we replace each entry in the genotype matrix $\mathbb{A}$ that used to contain symbols A/B with a probability. The probability $\mathbb{A}[i,j]$ represents the confidence that we have about marker l_i in individual c_j of being in state A. For the known observations the probabilities are fixed to be 1 or 0 depending whether the genotypes observed is A or B, respectively. The probabilities for the missing observations will be initially set to 0.5. Given that now $\mathbb{A}$ contains probabilities, the pairwise distance between two markers l_i and l_j can computed as follows

$$d_{i,j} = \sum_{1\leq k\leq n} \mathbb{A}[i,k](1-\mathbb{A}[j,k]) + (1-\mathbb{A}[i,k])\mathbb{A}[j,k] \tag{1}$$

Our iterative algorithm works as follows. First, given the input matrix $\mathbb{A}$, we compute the pairwise distance according to (1). Then, we run our MST-based algorithm to find the most probable order of the markers. This constitutes the M step in the EM framework. Given the new marker order, we can adjust the estimatate for a missing observation from marker i_2 on individual j as follows

$$\mathbb{A}[i_2,j] = \sum_{a\in\{\mathtt{A},\mathtt{B}\},c\in\{\mathtt{A},\mathtt{B}\}} L_{a,\mathtt{A},c} / \sum_{a\in\{\mathtt{A},\mathtt{B}\},b\in\{\mathtt{A},\mathtt{B}\},c\in\{\mathtt{A},\mathtt{B}\}} L_{a,b,c} \tag{2}$$

where i_1 is the marker immediately preceding i_2 in the most recent ordering, i_3 is the marker immediately following i_2, and $L_{a,b,c}$ is the likelihood of the event $(l_1 = a, l_2 = b, l_3 = c)$ at the three consecutive loci. The quantity $L_{a,b,c}$ is straighforward to compute. For example, $L_{\mathtt{A},\mathtt{A},\mathtt{A}} = \mathbb{A}[i_1, j](1-\hat{\mathbf{P}}_{i_1,i_2})(1-\hat{\mathbf{P}}_{i_2,i_3})\mathbb{A}[i_3, j]$, where $\hat{\mathbf{P}}_{i_1,i_2} = d_{i_1,i_2}/n$, $\hat{\mathbf{P}}_{i_2,i_3} = d_{i_2,i_3}/n$ are the MLEs for $\mathbf{P}_{i_1,i_2}$ and $\mathbf{P}_{i_2,i_3}$ respectively, and d_{i_1,i_2} and d_{i_2,i_3} are as computed according to (1) in the most recent M step. In the case where the missing observation is at the beginning or the end of the map, the above estimates will have to be adjusted slightly. The new estimation of the probabilities corresponds to the E step in the EM framework.

An E-step is followed by another M-step, and this iterative process continues until the marker order converges. In our experimental evaluations, the algorithm converges pretty quickly, usually in less than ten iterations. The pseudo-code of our algorithm, called MSTMAP, is presented in the Appendix.

5 Experimental Results

We implemented our algorithm in C++ and carried out extensive evaluations on both real data and simulated data.

The real data comes from our ongoing genetic mapping project for *Hordeum vulgare* (barley) at University of California, Riverside. In total there are three mapping populations being studied, all of which are DH populations. The first mapping population is the result of crossing Oregon Wolfe Barley Dominant with Oregon Wolfe Barley Recessive (see `http://barleyworld.org/oregonwolfe.php`), and from here on, we will refer to it as the OWB data set. The OWB data set consists of 1,020 markers genotyped on 93 individuals. The second mapping population is the result of a cross of Steptoe with Morex (see `http://wheat.pw.usda.gov/ggpages/SxM/`), which consists of 800 markers genotyped on 149 individuals. It will be referred to as SM data set from here on. The third mapping population is the result of a cross of Morex with Barke recently developed by Nils Stein and colleagues at the Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), which contains 1,068 markers on 93 individuals. This data set will be referred to as MB in our discussion. The genotypes of SNPs for the above data sets were determined via the Illumina GoldenGate Assay. These three barley data sets were expected to contain seven LGs, one for each of the seven barley chromosomes.

We generate the simulated data set according to the following procedure (which is the same as that used in [16]). First we decide how many markers to place on the genetic map, how many individuals to genotype, what is the error rate and what is the missing rate. We then produce a "skeleton" map, according to which the genotypes for the individuals will be generated. The markers on the skeleton map are spaced at a distance of 0.5 centimorgan plus a random distance according to a Poisson process. On average, the adjacent markers are 1 centimorgan apart from each other. We generate the genotypes for the individuals as follows. The genotype at the first marker is generated at random with probability 0.5 of being A and probability 0.5 of being B. The genotype at the next marker depends upon the genotype at the previous marker and the distance between them. If the distance between the current marker and the previous marker is x centimorgans, then with probability $x\%$, the genotype at the current locus is the opposite

Table 1. Summary of the clustering results for the barley data sets. $\bar{\rho}$ is the average ρ of the seven largest LGs in each population.

Data set	*# markers*	*# LGs*	*Sizes of the LGs*	$\bar{\rho}$
OWB	1,020	7	105,178,165,132,146,129,165	0.9948
SM	800	8	149,105,147,73,85,147,93,1	0.9972
MB	1,068	8	150,198,136,162,96,130,194,2	0.9950

of that at the previous locus, and with probability (1-$x\%$) the two genotypes are the same. Finally, according to the specified error rate and missing rate, we flip the current genotype to purposely introduce an error or simply delete it to introduce a missing observation. Following this procedure, we generated various datasets on a wide range of choices for n, m, error rate and missing rate.

5.1 Evaluation of the Clustering Algorithm

First, we evaluated the effectiveness of our clustering algorithm. We ran our clustering algorithm on the datasets for barley. Since the genome of barley consists of seven chromosome pairs, we expected the clustering algorithm to produce seven linkage groups. Using the default value for ϵ, our algorithm produced seven linkage groups for the OWB data set, and eight linkage groups for the MB and SM data set. The same results can be obtained in a rather wide range of values of ϵ. For example, for $\epsilon \in [0.000001, 0.0001]$ the OWB data set is always clustered into seven LGs. The smallest linkage group in the MB data set contains only two markers, whereas the smallest linkage group in the SM data set is a singleton. The explanation for these exceptions is not yet certain, but we conjectured some problems with the Illumina data. The result of the clustering algorithm is summarized in Table 1. We also compared the clusters produced by our algorithm against the ones produced by JOINMAP, which is a commercial software for linkage analysis. The clusters turned out to be identical. However, since JOINMAP implements a hierarchical clustering algorithm based on pairwise LOD score, the help of an expert is needed in order to decide where to cut in the dendrogram in order to produce the meaningful clusters. Our algorithm is conceptually much simpler.

5.2 Evaluation of the Quality of the Minimum Spanning Trees

Second, we verified that on real and simulated data, the MSTs produced by MSTMAP are indeed very close to TSP paths. This experimental evaluation corroborates the fact that the MST produces very good initial solution. Here, we computed the fraction ρ of the total number of bins/vertices in the graph that belong to the longest path (backbone) of the MST. The closer is ρ to 1, the closer is the MST to a path.

Table 1 shows that on the barley datasets, the average ρ for the seven linkage groups (not including the smallest ones in the SM and MB dataset) is always very close to 1. Indeed, 16 of the 21 MSTs are paths. The remaining 5 MSTs are all very close to paths, with just one node hanging off the backbone. When our algorithm generates MSTs which are paths we are guaranteed that they are optimal, thus increasing the confidence in the correctness of the order obtained.

On the simulated dataset with no genotyping errors, ρ is again close to one (see Figure 1-LEFT) for both $n = 100$ and $n = 200$ individuals. When the error rate is 1%, the ratio drops sharply to about 0.5. This is due to the fact that the average distance between nearby markers is only one centimorgan. One percent error introduces an additional distance of two centimorgans which is likely to move a marker around in its neighborhood. We computed ρ for several values of error rate, up to 15%. At 15% error rate, the backbone contains only about 1/4 of the markers. However, this short backbone is still very useful in obtaining a good map since it can be thought as a sample of the markers in their true order. Also, observe that increasing the number of individuals will slightly increase the length of the backbone. However, the ratio remains the same irrespective of the number of markers we include on a map (data not shown).

5.3 Evaluation of the Accuracy of the Ordering

In the third and final evaluation, we used the simulated data to compare the accuracy of the maps produced by MSTMAP against the ones generated by RECORD [16]. We selected RECORD because, to the best of our knowledge, it is the best tool available for genetic mapping. RECORD is a recent software tool which, according to the authors and based on our experience, outperforms JOINMAP. JOINMAP is another widely-used commercial software for linkage analysis. RECORD is also rather fast . The algorithm implemented in RECORD runs in time quadratic in the total number of markers, which is the same as in MSTMAP. Last but not least, RECORD is command line based, which allows us to run more extensive tests without too much human intervention. All the other tools we are aware of (e.g., CARTHAGENE, ANTMAP, JOINMAP) are GUI-based.

We used the Kendall's *concordance correlations* to evaluate the quality of the maps. Kendall's metric, denoted by τ, is commonly used in statistics to measure the correlation between two rankings. This metric is given by $\tau = \frac{4(\text{\# concordant pairs})}{m(m-1)} - 1$, where a pair of markers is *concordant* if the relative order between them is the same in the two maps (one is the true map and the other is the map generated by either MSTMAP or RECORD), where m is the total number of markers. The value of τ ranges from -1 to 1. $\tau = 1$ when the two maps are identical; $\tau = -1$ when one map is the reverse of the other. Since the orientation of the map is not important in this context, whenever τ is negative, we flip one of the map and we recompute τ. In our case, $\tau \in [0, 1]$. The higher the value of τ, the closer is the map produced to the true map. Note that τ is more sensitive to global reshuffle than local reshuffle. For example, assume that the true order is the identity permutation. τ for the following order $\frac{n}{2}, \frac{n}{2}+1, \frac{n}{2}+2, ..., n, 1, 2, 3, ..., \frac{n}{2}-1$ is 0, whereas τ for the order $2, 1, 4, 3, 6, 5, ..., n, n-1$ is $(1 - 2/n)$ which is still close to 1 when n is large. The fact that τ is more sensitive to global reshuffle is a desired property since biologists are more interested in the correctness of the global order of the markers than the local order.

The results of the evaluation based on τ for $n = 100$ individuals are summarized in Figures 1-RIGHT, 2-LEFT, and 2-RIGHT. Four observations are in order. First, Figure 1-RIGHT shows that when the error rate is low, both MSTMAP and RECORD perform equally well. However, when the error rate gets higher, MSTMAP consistently

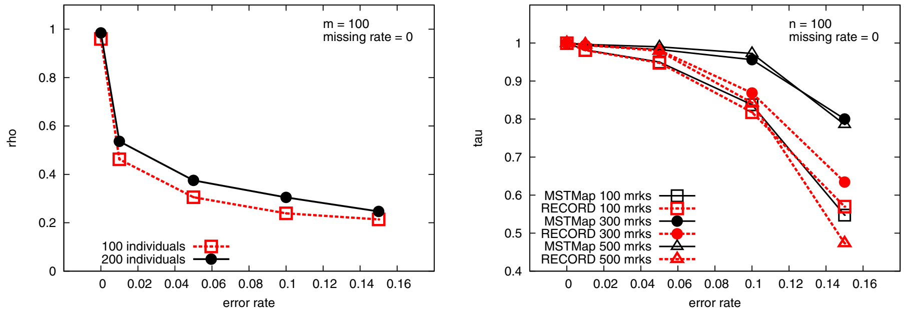

Fig. 1. Average ρ (LEFT) and τ (RIGHT) for thirty runs on simulated data for several choices of the error rates (and no missing data). n is the number of individuals, and m is the number of markers.

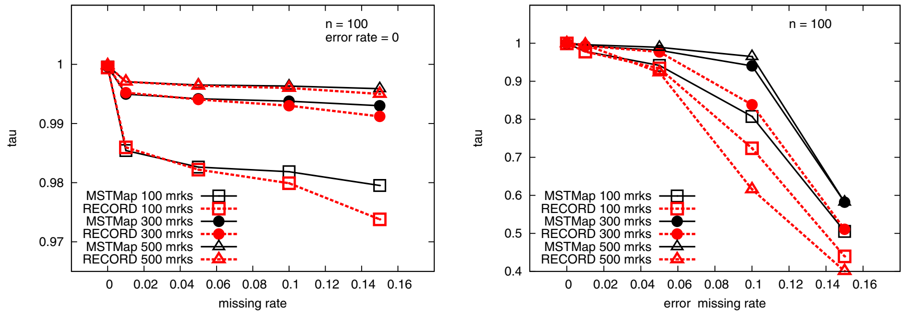

Fig. 2. Average τ for thirty runs on simulated data. LEFT: various choices of missing rates (error rate=0); RIGHT: various choices of error/missing rates (missing rate=error rate).

builds much more accurate maps than RECORD. Second, observe in Figure 2-LEFT that the missing data rate does not have too much of a negative impact on the quality of the maps assembled as the error rate. Even at the missing rate of 15%, the maps assembled are still very accurate (τ is larger than 0.99). Third, Figures 1-RIGHT and 2-RIGHT show that when the data is noisy, the performance of MSTMAP improves as the number of markers m increases. Forth, we observe that if number n of individuals increases, the quality of the maps constructed by both algorithms also improves (data not shown).

6 Conclusions

We presented a novel method to cluster and order genetic markers based on genotyping data. Our method is based on solid theoretical foundations, is computationally very efficient, handles gracefully missing observation, and performs as well as the best tool in the scientific literature. Additionally, in the presence of noisy data, our method clearly outperforms the other tools.

References

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. The MIT Press and McGraw-Hill Book Company, Cambridge (2001)
2. de Givry, S., Bouchez, M., Chabrier, P., Milan, D., Schiex, T.: CARTHAGENE: multipopulation integrated genetic and radiation hybrid mapping. Bioinformatics (2004)
3. Falk, C.T.: Preliminary ordering of multiple linked loci using pairwise linkage data. Genetic Epidemiology 9, 367–375 (1992)
4. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman and Company, New York (1979)
5. Gaspin, C., Schiex, T.: Genetic algorithms for genetic mapping. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.) AE 1997. LNCS, vol. 1363, pp. 145–155. Springer, Heidelberg (1998)
6. Glover, F.: Tabu search-part I. ORSA Journal on Computing 1, 190–206 (1989)
7. Glover, F.: Tabu search-part II. ORSA Journal on Computing 2, 4–31 (1990)
8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, Reading (January 1989)
9. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journ. Am. Stat. Ass. 58(301), 13–30 (1963)
10. Iwata, H., Ninomiya, S.: AntMap: constructing genetic linkage maps using an ant colony optimization algorithm. Breeding Science 56, 371–377 (2006)
11. Jansen, J., de Jong, A.G., van Ooijen, J.W.: Constructing dense genetic linkage maps. Theor. Appl. Genet. 102, 1113–1122 (2001)
12. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)
13. Lin, S., Kernighan, B.: An effective heuristic algorithm for the traveling sales man problem. Operation research 21, 498–516 (1973)
14. Liu, B.: The gene ordering problem: an analog of the traveling sales man problem. Plant Genome (1995)
15. Mester, D., Ronin, Y., Minkov, D., Nevo, E., Korol, A.: Constructing large-scale genetic maps using an evolutionary strategy algorithm. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.) AE 1997. LNCS, vol. 1363, pp. 145–155. Springer, Heidelberg (1998)
16. Os, H.V., Stam, P., Visser, R.G.F., Eck, H.J.V.: RECORD: a novel method for ordering loci on a genetic linkage map. Theor. Appl. Genet. 112, 30–40 (2005)
17. Schiex, T., Gaspin, C.: CARTHAGENE: Constructing and joining maximum likelihood genetic maps. In: ISMB, pp. 258–267 (1997)
18. Stam, P.: Construction of integrated genetic linkage maps by means of a new computer package: Joinmap. The Plant Journal 3, 739–744 (1993)
19. Sturtevant, A.H.: The linear arrangement of six sex-linked factors in drosophila, as shown by their mode of association. Journal of Experimental Zoology 14, 43–59 (1913)
20. Sun, Z., Wang, Z., Tu, J., Zhang, J., Yu, F., McVetty, P.B., Li, G.: An ultradense genetic recombination map for brassica napus, consisting of 13551 srap markers. Theor. Appl. Genet. (2007)
21. Weeks, D., Lange, K.: Preliminary ranking procedures for multilocus ordering. Genomics 1, 236–242 (1987)
22. Wilson, S.R.: A major simplification in the preliminary ordering of linked loci. Genetic Epidemiology 5, 75–80 (1988)