Genome analysis

**LPmerge: an R package for merging genetic maps by linear programming**

Jeffrey B. Endelman¹,*, Christophe Plomion²,³

¹Department of Horticulture, University of Wisconsin, Madison, WI 53706, USA
²INRA, UMR1202, BIOGECO, F-33610 Cestas, France
³Université de Bordeaux, UMR1202 BIOGECO, F-33170 Talence, France

**ABSTRACT**

**Summary:** Consensus genetic maps constructed from multiple populations are an important resource for both basic and applied research, including genome-wide association analysis, genome sequence assembly, and studies of evolution. The LPmerge software uses linear programming (LP) to efficiently minimize the mean absolute error between the consensus map and the linkage maps from each population. This minimization is performed subject to linear inequality constraints that ensure the ordering of the markers in the linkage maps is preserved. When marker order is inconsistent between linkage maps, a minimum set of ordinal constraints is deleted to resolve the conflicts.

**Availability:** LPmerge is on CRAN at http://cran.r-project.org/web/packages/LPmerge.

**Contact:** endelman@wisc.edu

**1 INTRODUCTION**

Broadly speaking, two types of strategies have been used to construct genetic maps across multiple populations. One is to minimize an objective function based on the observed recombination frequencies between markers, analogous to the strategy used for linkage mapping in a single population. Examples of this approach include the software packages JoinMap (Van Ooijen, 2006) and MultiPoint (Ronin et al., 2012). The second strategy is to work directly with the component linkage maps instead of the underlying recombination frequencies, which can lead to significant gains in computational efficiency without compromising map accuracy (Wenzl et al., 2006). This second strategy is employed by the software package MergeMap (Wu et al., 2011), which has been used in several different species (Muñoz-Amatriain et al., 2011; Khan et al., 2012; Gautami et al., 2012).

Endelman (2011) identified a weakness in MergeMap and proposed an alternative algorithm for merging linkage maps based on linear programming (LP), which was incorporated into an R package called DAGGER. The accuracy of the LP algorithm was validated on simulated data, and in a comparison with MergeMap using real barley data was found to have lower mean-squared error on a genome-wide basis (Endelman, 2011). A significant limitation of DAGGER was that it required the component linkage maps to have a consistent marker order. In practice the linkage maps from different populations often have ordering conflicts, even when there is a single physical order, because of genotyping errors and statistical errors arising from the use of small populations. Another weakness of DAGGER was that the length of the consensus genetic map was shrunken compared to the original linkage maps.

Both of these limitations have been overcome to create a new R package called LPmerge, which is available on CRAN. The LP algorithm has been reformulated without graph theory to eliminate shrinkage of the consensus map, and a new algorithm for resolving ordinal conflicts between linkage maps has been added. The objectives of this Note are to describe these modifications and to illustrate the performance of the conflict resolution algorithm.

**2 ALGORITHM**

**2.1 Consensus map error**

In the first step of LPmerge, markers are assigned to bins. If in every map where two markers were jointly mapped they co-segregated (had the same map position), they are placed in the same bin. Due to some markers co-segregating in one map but not in another, a single linkage map bin may be represented by multiple consensus map bins. Map positions in the i\(^{th}\) linkage map are denoted by y\(_i\), and consensus map positions are denoted by x. Within linkage map i, the markers are ordered from j = 1 to M\(_i\), and the map distance between the j\(^{th}\) and (j+q)\(^{th}\) markers is y\(_{i(j+q)}\)-y\(_{i(j)}\). Letting u\(_{i(j)}\) denote the consensus map bin containing marker j from map i, the corresponding distance in the consensus map for these two markers is x(u\(_{i(j+q)}\))−x(u\(_{i(j)}\)). The absolute error between the consensus map and the i\(^{th}\) linkage map for this interval is thus

\[
E_{i,j} = |x(u_{i(j+q)}))−x(u_{i(j)}))|−|y_{i(j+q)}−y_{i(j)}|
\]

The total error between the consensus map and the linkage maps is a sum over maps (i), markers (j), and interval sizes (q):

(1)
where $T$ is the number of linkage maps and $K$ is the maximum interval size. At the end of a linkage map, when the sum $j+q$ exceeds $M_i$, this expression is evaluated as if the linkage map were circular rather than linear. For example, when $j = M_i - 1$ and $q = 3$, the expression $j+q$ evaluates to 2. These “wrap-around” error terms keep the total consensus map length commensurate with the average linkage map length. The normalization factor

$$N_i = K \sum_{j=1}^{M_i}$$

is the number of error terms for map $i$, and $W_i$ are a set of possible weights for the average ($W_i = 1$ for unweighted). The maximum interval size $K$ can be varied to produce different consensus maps, and additional criteria can be used to select one. A tutorial illustrating this process is available at http://potatobreeding.cals.wisc.edu/software.

### 2.2 Resolving marker order conflicts

When minimizing the error (Eq. 2), it is desirable that the consensus map be as consistent as possible with the linkage maps in terms of marker order. This is achieved through the use of linear inequalities. For a pair of adjacent linkage map bins $(v,w)$, each with a single consensus map bin, the corresponding constraint is $x(w) - x(v) \geq 0$. When the linkage map bins contain multiple consensus map bins, constraints are added for every combination of the consensus map bins. The total set of ordinal constraints can be written in matrix notation as $A x \geq b$. If there are conflicts in marker order between the maps being merged, the linear system $A x \geq b$ will be infeasible. Finding the minimum number of constraints to remove to achieve feasibility is NP-hard (Amaldi and Kann, 1995). LPmerge uses a polynomial-time approximation from Chinneck (2001) (Algorithm 1), which is based on the idea of elasticizing constraints. The elastic LP corresponding to $A x \geq b$ is

$$\min \sum_{i,b} b_i$$

$$Ax + b \geq 1$$

$$b \geq 0$$

where $b$ are the elastic variables and the objective is to minimize their sum. The algorithm proceeds by finding which constraint (row of $A$) leads to the lowest elastic sum when removed. This constraint is then removed, and the procedure is repeated. When the algorithm eliminates a constraint and finds the elastic sum is zero, that subsystem is feasible and the algorithm stops.

### 2.3 LP problem

After removing the ordinal conflicts in $A$, linear programming is used to find a consensus map with minimum error (see supplementary material online).

### 3 RESULTS

Figure 1 illustrates the performance of the conflict resolution algorithm on a toy problem from the LPmerge reference manual. Four linkage maps have been merged (I–IV), each with seven markers (A–G). Map I represents the true marker order, and in the other maps the order of two adjacent markers has been inverted (highlighted in red) to represent mapping errors from the single population analyses. LPmerge correctly identifies the outliers from each map and eliminates them, as documented in the session log output:

- **Eliminated following constraints**
  - Map II: $C < B$
  - Map III: $G < F$
  - Map IV: $B < A$

Unlike other map-merging algorithms, LPmerge removes inequality constraints rather than markers to resolve conflicts. The consensus map returned by LPmerge for this toy problem has the correct marker order and distances (Fig. 1).

**REFERENCES**


