Selective Mapping: A Strategy for Optimizing the Construction of High-Density Linkage Maps

Todd J. Vision,* Daniel G. Brown,† David B. Shmoys,†,‡ Richard T. Durrett§ and Steven D. Tanksley*

**Department of Plant Breeding, †Department of Computer Science, §Department of Mathematics and ‡School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853*

Manuscript received June 21, 1999
Accepted for publication January 12, 2000

ABSTRACT

Historically, linkage mapping populations have consisted of large, randomly selected samples of progeny from a given pedigree or cell lines from a panel of radiation hybrids. We demonstrate that, to construct a map with high genome-wide marker density, it is neither necessary nor desirable to genotype all markers in every individual of a large mapping population. Instead, a reduced sample of individuals bearing complementary recombinational or radiation-induced breakpoints may be selected for genotyping subsequent markers from a large, but sparsely genotyped, mapping population. Choosing such a sample can be reduced to a discrete stochastic optimization problem for which the goal is a sample with breakpoints spaced evenly throughout the genome. We have developed several different methods for selecting such samples and have evaluated their performance on simulated and actual mapping populations, including the Lister and Dean *Arabidopsis thaliana* recombinant inbred population and the GeneBridge 4 human radiation hybrid panel. Our methods quickly and consistently find much-reduced samples with map resolution approaching that of the larger populations from which they are derived. This approach, which we have termed *selective mapping*, can facilitate the production of high-quality, high-density genome-wide linkage maps.

SINCE its inception in the early decades of this century, genetic linkage mapping has been based on random sampling of individuals from large recombinant populations (F_2's, doubled haploids, *etc.*) and, more recently, from panels of radiation hybrid cell lines. A random sampling approach provides a means of mapping with little prior knowledge about each individual. However, large samples of recombinational crossover sites or radiation-induced fragmentation sites, collectively referred to as *breakpoints*, can only be obtained by genotyping large numbers of individuals.

Since individuals differ in the number and distribution of breakpoints, different combinations of individuals complement one another to varying degrees in the order and position information they provide. In principle, with prior knowledge of the number and position of breakpoints in a mapping population, it should be possible to select a sample of individuals with a more desirable distribution of breakpoint sites than is likely in a random sample of the same size. (We discuss some possible measures of "desirability" in what follows.) Given the magnitude of current research efforts in the area of linkage mapping (*e.g.*, Wang *et al.* 1998), and the potential savings in throughput and expense to be gained from culling relatively uninformative genotypes, a selective sampling approach is highly desirable. To that end, we have developed computational methods for finding good mapping samples and have tested the application of these methods to several widely used mapping populations, including a set of *Arabidopsis thaliana* recombinant inbred lines (Lister and Dean 1993) and the GeneBridge 4 human radiation hybrid panel (Gyapay *et al.* 1996).

A two-phase mapping approach: We propose that there be two experimental phases in the construction of a high-density genetic map: the first is to construct a high-confidence framework and the second is to add new markers to this framework. This two-phased strategy allows many markers to be placed on a well-measured map with a minimum of genotyping and avoids the loss in map resolution that would result from arbitrarily shrinking mapping population size. We have dubbed this strategy *selective mapping*. Similar strategies have been proposed for determining the linkage relationships between markers and phenotypic variants (*e.g.*, Paterson *et al.* 1991; Darvasi 1997) and selective genotyping of human pedigrees has been employed to map markers already known to be in a particular region (Fain *et al.* 1996). However, we are aware of no previous application of these ideas to whole-genome mapping of molecular markers.

In the first phase of the proposed strategy, the breakpoints for each individual in the full mapping popula-

Corresponding author: Todd Vision, USDA-ARS Center for Bioinformatics and Comparative Genomics, 604 Rhodes Hall, Cornell University, Ithaca, NY 14853. E-mail: tv23@cornell.edu

tion are located using a limited number of the available markers, which we refer to as the *framework markers.* Preferably, these markers are chosen on the basis of prior knowledge concerning their even distribution throughout the genome, as measured by breakpoint density. The map constructed in this first phase, in which the framework markers are placed confidently and precisely, is referred to as the *framework map.* This concept has been modified somewhat from that of Keats *et al.* (1991). In the second phase, the genotypes for all subsequent markers are scored in a small sample of individuals that have been selected on the basis of the information obtained during the first phase. The data obtained in this second phase allow the mapping of new markers relative to the fixed framework. We have recently developed methods of analysis designed specifically to accommodate selected samples (D. G. Brown and T. J. Vision, unpublished results). Conventional mapping software packages may also be modified for this task.

Map resolution: A major concern raised by selective mapping is how much of a cost one must pay in map resolution in return for the benefit of genotyping markers on only a subset of the mapping population. To answer this question, and to explore ways of minimizing this sacrifice, it is necessary to precisely define map resolution.

We first define a *bin* to be an interval along a linkage group within which no breakpoints occur among any members of a given set of individuals but which is bounded by such breakpoints in at least one individual (or by the end of a linkage group; see Figure 1). Bins are the smallest unit of resolution in a genetic map; two or more loci within a single bin cannot be ordered relative to one another without supplementary information. This limit to resolution is a very real problem for a high-density map. Ben-Dor and Chor (1997) note that it is currently impractical to construct a mapping population with sufficient breakpoint density, even for a radiation hybrid panel, that the relative order may be resolved for every triplet of linked markers when the number of markers is much greater than 100.

We define the *map resolution* for a given set of individuals to be the set of the lengths of the bins in that set of individuals. Assuming that the location of each breakpoint is unique, and that no more than one population member has zero breakpoints, different samples of the population will also differ in the distribution of bin boundaries and bin lengths. By selecting a collection of individuals with optimal (or near optimal) map resolution for a given size, we aim to place markers with greater precision in the second phase of selective mapping than would likely happen using a random sample of the same size from the same mapping population.

Evaluation statistics: The observed distribution of bin lengths, and thus the resolution of a genetic map, may be characterized by many different evaluation statistics. Three measures that we discuss here are the average bin length (ABL), the sum of the squares of the bin lengths (SSBL), and the maximum bin length (MBL).

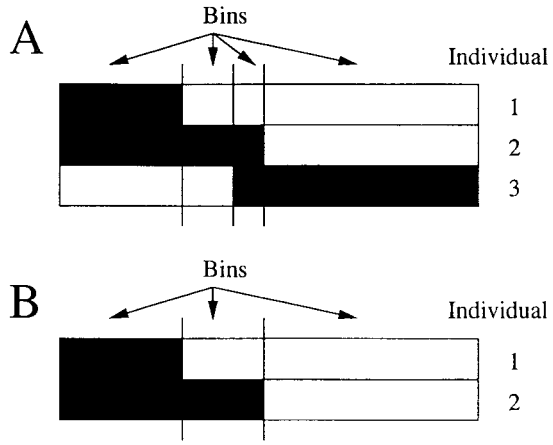

Figure 1.—The concept of a bin. A bin is the interval between the most closely adjacent breakpoints in the sample under consideration. This schematic shows that while deleting one or more individuals may result in there being fewer bins, this need not result in an increased maximum bin length. Boundaries between shaded and unshaded areas represent breakpoints. Bin lengths are drawn to scale. Individuals are assumed to be haploid. (A) The inclusion of all three individuals breaks the interval into four unequal length bins. (B) Removing the third individual causes the loss of the smallest bin but the maximum bin length remains unchanged.

The ABL is easily minimized. It is equal to the sum of all bin lengths (the genome length) divided by the number of bins. Since the first of these is constant, the function is minimized by maximizing the number of bins. Thus, the sample of k individuals out of a mapping population of size n that minimizes the ABL consists of the k individuals with the most breakpoints, assuming all breakpoints are at unique positions. Unfortunately, this sample may contain a small number of very long bins, resulting in a considerable fraction of markers being coarsely mapped. Since this is undesirable, we suggest that it is preferable to minimize one of the alternative statistics: either the SSBL or the MBL.

Minimizing the SSBL is equivalent to minimizing the expected length of the bin containing a marker chosen uniformly at random from the whole genome. To see this, consider a genome of length L, divided into bins of length $l_1, \ldots l_m$. A uniformly chosen marker has probability $p_1 = l_1/L$ to be in the first bin, $p_2 = l_2/L$ to be in the second, and so on. The expected length of the bin containing a uniformly chosen marker is $\Sigma_{i=1}^{m} p_i l_i = (1/L)\,\Sigma_{i=1}^{m} l_i^2$. Since L is constant, the expected bin length is directly proportional to the SSBL. We note that minimizing the SSBL is not the same as minimizing the variance of the bin lengths. While the two functions are similar, the SSBL implicitly takes into account the number of bins. It can be thought of as a single statistic incorporating information about both the variance and the mean of the bin length distribution.

The MBL provides an upper bound to the length of

the longest bin in the genome, which may be an attractive prospect to some investigators. On the other hand, we have found that minimizing SSBL maximizes the genome-wide accuracy and precision in marker placement (D. G. Brown and T. J. Vision, unpublished results). We feel that both the MBL and the SSBL are legitimate as minimization goals. Furthermore, they are closely correlated among samples.

METHODS AND RESULTS

We first consider an idealized case, in which the locations of the breakpoints are exactly known. We are given a population P with n members, which we label as $P = \{1, 2, \ldots, n\}$, and seek the best sample subset of the population for a given size k.

We cannot consider all $\binom{n}{k}$ possible samples, since this number is prohibitively large for realistic values of n [*e.g.*, $\binom{100}{30} = 10^{26}$]. Instead, we have considered a number of naive heuristics, which turn out to perform rather poorly, and have developed much more desirable algorithms using ideas from the field of discrete optimization.

To evaluate the quality of a possible sample, we order all of the breakpoints in the members of the sample and compute the MBL, that is, the longest distance between consecutive breakpoints (including the ends of linkage groups). This statistic is our *objective function*; for each sample size k, we seek the population sample of that size that minimizes the objective function. Later, we consider data from real populations, where we do not know the exact sites of the breakpoints. For such cases, we will seek a sample that minimizes a slightly modified objective function.

Let the *performance ratio* of a sample be the ratio of the objective function value for the sample to the objective function value for the population as a whole. Clearly, the best possible performance ratio is 1.0, and the performance ratios of all algorithms will approach 1.0 from above as k approaches n.

Naive algorithms: One naive algorithm is to generate a large number of random samples of size k and select the generated sample with minimum objective function value. A less naive algorithm is to choose the sample that consists of the k individuals having the most breakpoints. As discussed above, this latter sample is guaranteed to have the smallest possible ABL for a given k. However, it may be far from optimal at minimizing the MBL or SSBL.

Preferred algorithms: In addition to evaluating the performance of these naive algorithms, we have made use of more sophisticated tools, including greedy algorithms, integer programming, and linear programming with randomized rounding. Neither greedy algorithms nor mathematical programming are new to biological applications. Greedy algorithms have been used for DNA sequence assembly (*e.g.*, Staden 1979), while mathematical programming has been used to design potential functions for protein folding (*e.g.*, Crippen 1991) and in the design of gene expression experiments (Karp *et al.* 1999), among other applications. In addition to these tools, we have also evaluated the utility of a clean-up procedure designed to ensure that each member of the chosen sample contributes to the value of the objective function.

Greedy algorithms: Greedy algorithms tend to be fast and to give satisfactory solutions, but can only guarantee to find a local optimum. The simplest formulation of the greedy algorithm is as follows. To construct a sample of size k, start with an empty sample. Then, until the desired sample size is reached, find the unchosen population member that, when added to the current sample, most improves our objective function. The underlying hope is that these k good choices will combine to give a sample that is good overall (Nemhauser and Wolsey 1988, p. 393). An alternative strategy is to start with the entire population as the sample, and at each step, remove the member from it that would have the least effect on the objective function, until the sample is of the desired size. We did not employ this strategy because it is computationally very expensive for large populations.

One can avoid focusing on a single local optimum by incorporating a limited element of randomness into the choice of each member of the sample set. To implement a randomized greedy algorithm, choose the next sample member uniformly at random from the r choices that would most improve the objective function for some small value of r (*e.g.*, 3 or 5; Resende 1998). Repeat this a large number of times, and choose the best sample set found.

To further improve upon this scheme, we have considered a *mixed greedy algorithm* that sequentially employs two different objective functions. To build a mixed greedy sample of size k, first build a sample S_1 of size $k/2$ by performing the greedy algorithm with the SSBL objective function. After $k/2$ selections, switch to the greedy algorithm to minimize objective function MBL, augmenting S_1 until a sample S of size k is obtained. The reasoning behind this approach is that minimizing the first objective function forces all bins to be small, rather than myopically focusing on the largest bin. Thus, upon switching to the MBL objective function, one has a better starting point than if one had begun with the MBL objective function initially. We have considered a randomized version of this algorithm where the half sample S_1 is produced by a randomized greedy procedure and the augmentation to the full sample is by a nonrandomized greedy algorithm. Since we have found that this randomized mixed greedy algorithm consistently outperforms simpler (nonrandomized or nonmixed) greedy algorithms both in simulations and in real data, we only report results for the randomized mixed greedy algorithm in our results on the MBL. We also report results using an unmixed randomized greedy algorithm to optimize SSBL alone.

Integer linear programming: Integer linear programming is an alternative solution method that has the advantage of finding the *optimal* sample of size k. However, it requires a prohibitive amount of computation for a large population.

Let the objective function be MBL and consider the population $P = \{1, \ldots, n\}$. To each member i of P, assign a *decision variable* y_i, where $y_i = 1$ if member i is included in our sample, and $y_i = 0$ otherwise. The constraints on the variables model the requirements of the sample and must be linear constraints in the decision variables. The first constraint on the y_i's is that $\Sigma_i y_i = k$, requiring that a sample of size k must be chosen. To formulate the second set of constraints, consider an objective function value B and suppose that it is possible to find a sample of size k that achieves this value. In this sample, the longest distance between consecutive breakpoints is less than or equal to B. To encode this into a finite number of constraints, we note that the distance between any breakpoint in the entire population and the next breakpoint in the sample is less than or equal to B, since each unchosen breakpoint is between two chosen breakpoints, which we know are separated by a distance less than or equal to B. For each breakpoint j in any of the members of the population, let $C_{j,B}$ be the set of population members i that have a breakpoint after j and within distance B of it. Then, for each breakpoint j in the entire population, at least one member of $C_{j,B}$ must have been chosen. For each breakpoint j, this requirement can be modeled by adding to the integer program the constraint $\Sigma_{i \in C_{j,B}} y_i \geq 1$. Thus, there is a sample of size k with objective function value less than or equal to B exactly when there is a solution to the following set of constraints, which is the *integer program corresponding to distance* B, or IP_B:

$$\sum_i y_i = k; \tag{1}$$

$$\sum_{i \in C_{j,B}} y_i \geq 1, \quad \text{for each breakpoint } j; \tag{2}$$

$$y_i \in \{0, 1\}, \quad \text{for each } i = 1, \ldots, n. \tag{3}$$

We say that a set of constraints is *feasible* if it can be satisfied by an assignment to the y_i decision variables (called a *feasible solution*); otherwise it is *infeasible*. One seeks B^*, the minimum possible value of B for which IP_B is feasible; this is the best objective function value that can be attained for sample size k. An optimal sample, then, consists of the k population members for which $y_i = 1$ is a feasible solution to IP_B. Note that there may be multiple feasible solutions for a given feasible set of constraints.

One can obtain the value of B^* solving a limited number of integer programs using a bisection search strategy. To do this, one maintains a range of values in which B^* may fall and cuts the range in half at each step based on the feasibility or infeasibilty of the integer program corresponding to the midpoint of the range (Nemhauser and Wolsey 1988, p. 127).

Unfortunately, integer programs can be quite slow to solve, and their solution time increases quite dramatically as their size increases (Nemhauser and Wolsey 1988, p. 125). In this case, the solution time depends primarily on the population size. While the optimal threshold, and its associated sample, can be calculated for reasonably small, simulated data, this is not so for moderate to large mapping populations. Thus, we have employed integer programming primarily to evaluate the performance of alternative algorithms, none of which can guarantee an optimal sample.

Linear programming with randomized rounding: Linear programming, another form of mathematical programming, provides a common way to work around the computational intractability of integer programming. One "relaxes" the integer programming requirement that the y_i variables are 0 or 1 and simply requires that the y_i be in the closed interval from 0 to 1. One then obtains solutions that may fractionally choose population members, with the sum of the fractions still equal to k. For a given threshold B, we consider the following linear program, which we call LP_B:

$$\sum_i y_i = k; \tag{4}$$

$$\sum_{i \in C_{j,B}} y_i \geq 1, \quad \text{for each breakpoint } j; \tag{5}$$

$$0 \leq y_i \leq 1, \quad \text{for all } i = 1 \ldots n. \tag{6}$$

These linear programs, without the integer requirement on the y_i, are much easier to solve (Chvátal 1983) and assignments to the y_i variables that satisfy these constraints are still valuable despite being potentially fractional. For example, they allow us to judge the quality of a given sample. Let B^*_{LP} be the smallest value of B for which LP_B is feasible. Then $B^*_{LP} \leq B^*$, since if y is the 0–1 assignment that shows that IP_{B^*}. is feasible, then y is also a feasible assignment for LP_{B^*}. Suppose, then, we find a sample S with objective function value B_S, which is close to B^*_{LP}. Then $B^* \leq B_S$, since B^* is the best possible value. So $B^*_{LP} \leq B^* \leq B_S$, and if B_S is close to B^*_{LP}, then it must also be close to B^*. Hence, the linear program gives a lower bound on the optimal objective function for the integer program.

In addition to obtaining lower bounds on B^* with the linear programs, one can also use feasible y_i assignments to the linear programs to select actual samples. Each of the y_i will range from zero to one. If we treat y_i as the probability with which we choose to include population member i, and we make these choices independently, then the expected sample size is k. Intuitively, this "randomized rounding" scheme takes advantage of the information in the y_i. If y_i is near one, one is likely to include member i; if it is near zero, it will probably not be included in our sample (Raghavan and Thompson 1987). If the sample chosen is smaller than k, augment

it greedily until it is of size *k*; if too large, remove the sample members whose deletion has the least impact on the objective function until the sample is of size *k*. Repeat the process many times and choose the best sample set found.

Integer and linear programs designed to minimize SSBL, as opposed to MBL, are too computationally intensive to be practical even for moderately sized data sets. Therefore, we report only mathematical programming results for the MBL objective function.

Post-selection clean-up: In optimizing the MBL, it is often possible to improve the samples chosen by the greedy and randomized rounding algorithms described above by adding a *clean-up routine*. The longest interbreakpoint interval of a given sample is defined by only two sample members: the member that includes the first breakpoint of that interval and the member that includes the second breakpoint. Taking advantage of this fact, implement a clean-up routine by looking at a provisionally selected sample, in the order in which the individuals were added, and removing individuals whose deletion does not increase the maximum bin length. Then restart the greedy algorithm and augment the now incomplete sample until it reaches size *k*. Repeat this process until removing *any* single member from the sample would increase the MBL. In the experiments we report below, the randomized mixed greedy and randomized rounding algorithms both include this clean-up routine.

Simulated populations with exactly specified breakpoints

We evaluated the performance of our sample selection algorithms on simulated data with exactly specified breakpoints. We looked for the minimum sample sizes at which these algorithms found samples whose objective function values were close to the objective function value for the full population. We then chose samples of constant size from populations of increasing size. This was done with the twofold aim of evaluating the sensitivity of algorithm performance to population size and providing some guidance as to how large a population size should be used in constructing a framework map. All of our test codes were implemented in Matlab 5.3 (Mathworks, Natick, MA) and run on a Sun Ultra Sparc 10 workstation. Our linear and integer programs were solved with CPLEX 4.0 (CPLEX Optimization, Incline Village, NV), an industrial mathematical programming package.

Fixed population size with increasing sample size: In our first set of tests, we simulated 10 F_1 recombinant populations of 100 haploid organisms with a genome length of 1000 cM. Breakpoint sites were generated by simulating, for each organism, a homogeneous Poisson process with mean inter-breakpoint distance of 100 cM. For all sample sizes, the randomized greedy algorithms ran in less than 10 sec. The computation time for the randomized rounding procedure was typically less than 5 min and was almost entirely devoted to solving the linear programs.

The results are shown in Figure 2A. The figure shows, for several algorithms, the mean performance ratio (the MBL of the sample divided by the MBL of the entire population) for the samples found of a given size. We also computed the optimal samples of each size using integer programming. The randomized mixed greedy and linear programming with randomized rounding solutions were nearly indistinguishable from one another and were extremely close in MBL to the optimal samples found by integer programming.

On the other hand, the two naive algorithms (choosing the best of 50 random samples and choosing the sample containing the most breakpoints) required samples approximately twice as large, or larger, than would be optimal for a given performance ratio. For example, the mean performance ratio of the samples from the randomized mixed greedy algorithm of size 25 was 1.27; this performance ratio was not achieved until size 52 when the best of 50 random samples was selected and until size 68 when the population members with the most breakpoints were selected. We examined whether choosing the best out of a larger set of random samples (1000, instead of 50) made a qualitative difference to this conclusion and found that it did not (results not shown). Figure 2A also shows the average, as opposed to the best, performance ratio among 100 random samples. If a researcher were to randomly cull a population comparable to this one, for whatever reason, this is the sacrifice in mapping resolution that would result. As can be seen from the comparison of the average and the best random samples, the variability among the random samples was small compared to the distance between these samples and those selected by the more sophisticated algorithms.

The addition of the clean-up routine made a substantial contribution to the quality of the samples chosen, particularly for the randomized rounding routine. The cleaned randomized rounding samples had a performance ratio 12.6% closer to 1.0 than uncleaned samples, when samples of all sizes were considered. We also found that, in the absence of the clean-up routine, samples chosen by randomized rounding had significantly higher MBL than samples chosen by the randomized mixed greedy algorithm. When the clean-up routine was added, the performance ratios of samples chosen by these two algorithms became nearly indistinguishable. While the clean-up routine was appended to the end of these two algorithms and not the others, we found that this difference was not wholly responsible for the striking separation between the two classes of algorithms; the randomized rounding and randomized mixed greedy selected samples had consistently lower performance ratios than samples chosen by the other algorithms even in the absence of a clean-up routine (results not shown).

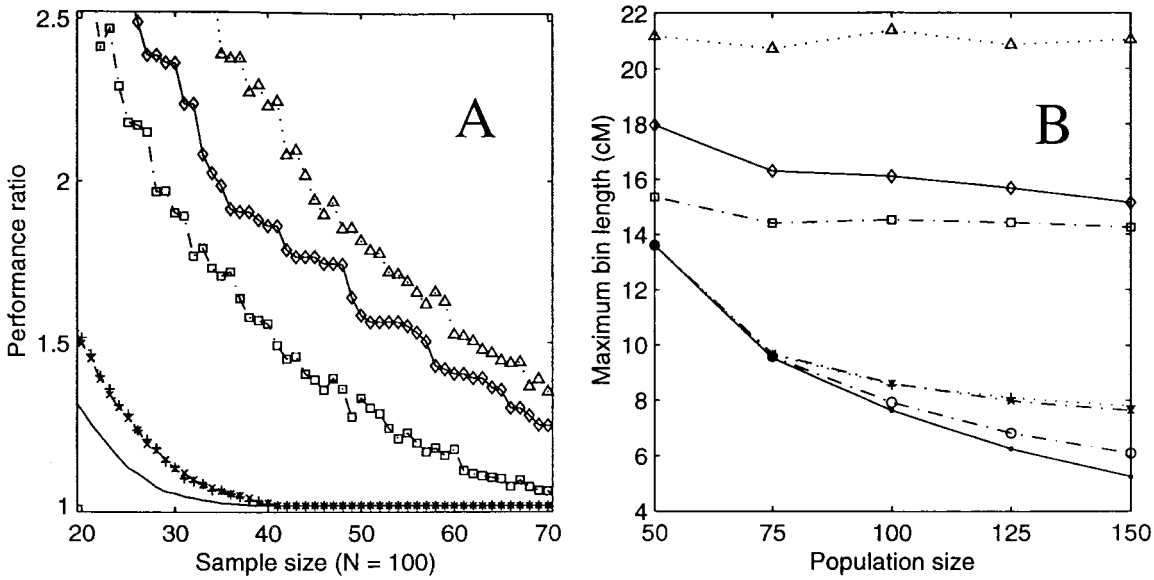

Figure 2.—Optimization for the maximum bin length in simulated populations with exact breakpoint placement. The haploid F_1 population size is 100 and the simulated map length is 1000 cM. (A) Shown are the average performance ratios, over 10 replicates, for samples of varying sizes chosen by each of six different algorithms. The performance ratio is defined as the ratio of the maximum bin length in the selected sample relative to that for the population as a whole. (B) Shown are the maximum bin lengths in samples of size 30 chosen from populations of variable size using several different algorithms. Plotted is the maximum bin length value averaged over 50 simulated populations. Integer programming results are not shown. (×) Random-mixed greedy; (+) randomized rounding; (◇) most breakpoints; (□) best of 50 random samples; (△) average random sample; (—, in A) the integer programming, or optimal, solution; (○, in B) the linear programming lower bound; and (●, in B) the whole population. Note that the random-mixed greedy and randomized rounding solutions approach the optimal solution (in A) or the lower bound (in B) and greatly outperform the alternative heuristic algorithms.

Increasing population size with fixed sample size: In the second set of simulations, we examined the effect of increasing the size of the mapping population while holding the sample size fixed. Better samples potentially exist within larger populations but, because the search space expands rapidly with increasing population size, it might be more challenging to find them. Furthermore, the map resolution of the best sample of a given size may not increase as rapidly as that of the whole population. Thus, we desired to measure both the absolute and relative map resolution as a function of population size.

We simulated 50 haploid recombinant F_1 populations each of sizes 50, 75, 100, 125, and 150, with genome length 1000 cM, as before, and generated samples of size 30 from each of these populations. The results are shown in Figure 2B; the figure shows, for each population size, the MBL in samples generated by each of the algorithms. Due to the computational speed of the randomized greedy algorithm, we were able to select samples of size 200 from populations as large as 500 in only a moderate amount of computer time. The linear programming algorithm was practical for populations of fewer than about 300. The integer program did not converge within 24 hr for populations of size 125 and 150, so these results were not obtained; for clarity, the integer programming results are not shown in Figure 2B.

At all of the population sizes considered, the randomized mixed greedy and randomized rounding algorithms performed quite well. In contrast, the best-of-50-random-sample algorithms selected samples of equally poor quality at all population sizes. Interestingly, the algorithm that chose the population members with the most breakpoints improved only slightly in performance as the population size increased.

The randomized rounding and randomized mixed greedy algorithms improved in an absolute sense as the population size increased; however, their performance ratios slowly increased, as well. In other words, the MBL of the whole population decreased faster than the MBL of selected samples of fixed size. The performance ratio of the linear programming lower bound B^*_{LP} also increased (Figure 2B), suggesting that the randomized rounding and randomized mixed greedy samples were still close to optimal for their size.

The MBL of the population as a whole approximately halved as the population size doubled. (This can be explained by noting that the entire population of N individuals is equivalent to a sample from a single homogeneous Poisson process with mean inter-event distance of $100/N$ cM). An important consequence of this nonlinear relationship between population size and map resolution is that, even in the absence of selected sampling, improvements in map resolution become increasingly more modest as the population size grows.

Optimizing the SSBL for simulated populations: We

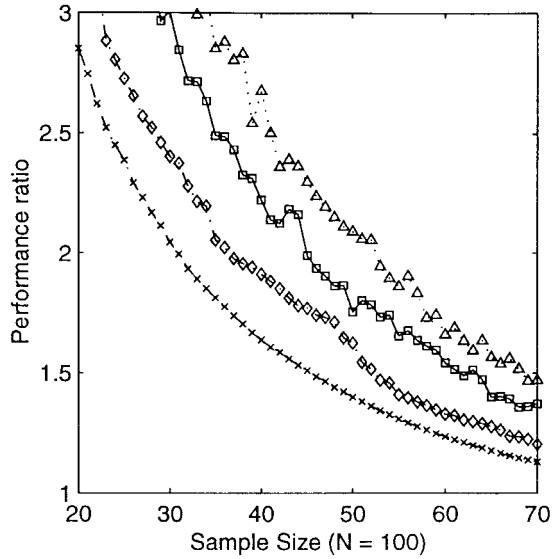

Figure 3.—Optimization for the sum of squares of bin lengths in simulated populations with exact breakpoint placement. The haploid F_1 population size is 100 and the simulated map length is 1000 cM. Shown are the performance ratios for one replicate, in samples of varying size, chosen by four different algorithms. The performance ratio is defined as in Figure 2. (×) Random greedy; (◇) most breakpoints; (□) best of 50 random samples; (△) average random sample.

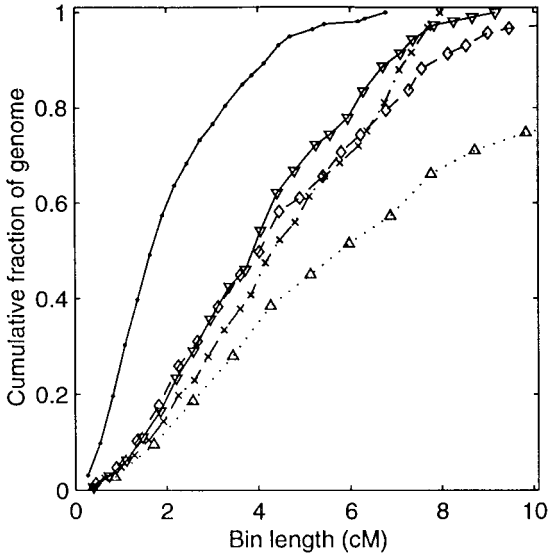

Figure 4.—Cumulative bin length distributions under different sample selection algorithms. The haploid F_1 population size is 100 and the simulated map length is 1000 cM. The fraction of the genome found in bins of less than a given bin length is shown for samples of size 30 found using four different sample selection methods and for the whole population. (●) Whole population; (▽) random greedy (minimizing sum of squares of bin lengths); (×) random mixed greedy (minimizing maximum bin length); (◇) most breakpoints (most breakpoints (minimizing average bin length); (△) single random sample.

also evaluated samples chosen solely to minimize SSBL in a 100-member simulated population. Figure 3 shows the result from this experiment, plotting the SSBL *vs.* the mapping population size for the best of 50 randomly chosen samples of each size, the sample chosen to have the most breakpoints, and the sample found by the greedy randomized algorithm. (A mathematical programming formulation is computationally impractical with this objective function.) The differences in sample quality among the algorithms are not as pronounced as for MBL, but it is still clear that the randomized greedy samples are of higher quality than can be found by naive methods.

We were also interested in the distribution of the bin lengths in samples chosen to minimize ABL, MBL, or SSBL. Figure 4 shows the cumulative fraction of the genome found in bins of various lengths for selected samples of size 30 chosen from the same 100-member simulated population. For reference, the distribution for the entire population is shown. Despite there being many small bins in the sample containing the most breakpoints, much of the genome is still represented by large bins. The samples minimizing MBL and SSBL concentrate more of the genome in small to moderate-length bins. Unlike the MBL sample, the SSBL sample does not accumulate bins that are just shy of the maximum length. On the other hand, the SSBL sample does have a slightly longer maximum bin length than the MBL sample. The ABL, MBL, and SSBL bin length distributions are clearly set off from that of the whole population at one extreme and that of a single random sample at the other.

Application to existing mapping populations: We applied these algorithms to a number of different existing mapping populations. We report on these results in detail for two populations constructed in very different ways. The first of these is a recombinant inbred (RI) population derived from a cross between *A. thaliana* ecotypes Columbia and Landsberg *erecta* (Lister and Dean 1993; http://nasc.nott.ac.uk/RIdata). We analyzed 101 lines scored for 261 of the identified framework markers spaced at an average of 2.0 cM apart. The total map length in this population is 513.1 cM. Since there is a twofold expansion of crossover frequency as a result of the repeated selfing of these lines (Haldane and Waddington 1931), the Arabidopsis RI population is closely comparable to the simulated haploid recombinant F_1 population, for which the total map length was 1000 cM. The second mapping population that we consider in detail is the GeneBridge 4 radiation hybrid (RH) panel of 93 hamster cell lines, each retaining about 32% of the human genome (Gyapay *et al.* 1996). Due to several large gaps between the framework markers in the RH map, we divided the linkage groups with gaps of length greater than 25 centirays (cR), or a population breakage frequency of 25%, into linkage groups within which all framework intervals were less than or equal to 25 cR. This, in principle, makes bin length

estimation more precise. However, it does not preclude the occurrence of bins greater than 25 cR since bins may span multiple framework intervals. We found in simulation experiments (results not shown) that introducing a small number of long (greater than 40 cM) gaps into a simulated framework map had only a small effect on the ability of our algorithms to find samples with small MBL. But there were many such gaps in the human data set, and we chose to be conservative. In total, we analyzed 55 human linkage groups with a map length of 10,866 cR. In addition to these, we analyzed a number of other published mapping data sets.

Unlike the simulated populations, in which breakpoints could be precisely localized, the marker genotypes in the real data sets allow us only to identify those intervals bearing odd numbers of breakpoints. Although unobserved breakpoints undoubtedly occur in the mapping populations under consideration, they do not contribute to our choice of a sample. For each observable breakpoint, we assume that there is exactly one breakpoint uniformly distributed between the flanking markers and that no breakpoints occur in intervals having identical flanking markers. Further, we assume that the sites of breakpoints are unique and independent of one another.

This conforms to our earlier assumption that breakpoint sites are generated by independent Poisson processes. It also assumes that map distances are additive, which is ideally the case. Since nonadditivity of map distances, and the related phenomenon whereby the map length expands with the addition of new markers, are largely due to the use of an inappropriate mapping function (Liu 1998) and/or the accumulation of genotyping errors (Lincoln and Lander 1992), sample selection should be preceded by careful error checking and rigorous analysis of framework marker data.

Updating the algorithms: Given the uncertainty in precise breakpoint location, we wish to minimize the *expectation* of the MBL or SSBL, under the assumption that known breakpoints are uniformly and independently distributed within framework intervals. A closed-form formula for E (MBL) is not readily available. So, to evaluate the MBL objective function, we generate 100 replicates of the population in which all breakpoints have been instantiated (*i.e.*, randomly resolved to exact sites). We compute the mean quality of our chosen sample for these replicates using the same objective function as in the simulations described above, where breakpoints were known with precision. For E(SSBL), we have derived a closed-form solution given known marker sites and a known number of breakpoints between consecutive markers. For a full derivation of this formula, see appendix a. Accordingly, our mixed greedy algorithm selects the first half of the sample without the need to randomly resolve the breakpoint locations many times over, thereby improving both the accuracy and the speed of the algorithm.

The mathematical programming algorithms require further modification to handle the imprecisely specified breakpoints of the RI and RH data. For the randomized rounding algorithm, we treat a feasible set of y_i variables to a very closely related linear program as the probability that each population member is either rejected or accepted to the sample and then greedily adding to the sample until it is of full size. See appendix b for the differences between the linear program solved in the exact case and that needed for the stochastic case. Integer programming is not appropriate in this case, as our previous integer programming model assumed knowledge of exact breakpoint placement. We note that the linear programs solved in this case are much smaller than for the exact case and can easily be solved for populations of size greater than 300.

Results for Arabidopsis, human, and other data sets

For the Arabidopsis data (Figure 5A), we found that the randomized mixed greedy and randomized rounding algorithms both performed well, although not as well as on simulated data, and they greatly outperformed the naive algorithms. For example, the randomized mixed greedy algorithm found a sample of size 30 with performance ratio less than 1.3; to achieve the same performance ratio with a sample containing those lines with the most breakpoints required 45 individuals. For samples of size 30, the randomized mixed greedy and randomized rounding algorithms ran in less than 5 min. Much of this time was occupied in repeated evaluations of the objective function via simulation during the greedy additions.

For the radiation hybrid data (Figure 5B), we found that both the differences among the algorithms and the improvements obtained by selective sampling were less dramatic than for the recombinant inbred data. While the shape of the relationship between performance ratio and minimum sample size was similar for both data sets, the samples required to achieve a given performance ratio were approximately 50% larger for the human data. In addition, the superiority of the randomized mixed greedy and randomized rounding algorithms diminished as the performance ratio approached 1.0. Still, for modest sizes, we did experience significant improvements. A sample of size 47 was obtained by both the greedy and linear programming algorithms with a performance ratio less than 1.5. By comparison, the same-sized sample containing those cell lines with the most visible breakpoints had a performance ratio greater than 1.7.

The superior performance of selective sampling in the Arabidopsis population appears to be due to the smaller number of breakpoints per individual. Comparisons of simulated data with comparable average breakpoint densities (10 or 100 per individual) gave qualitatively the same results as the Arabidopsis and human data, respectively (results not shown). That is, for the higher breakpoint density, significantly larger sample

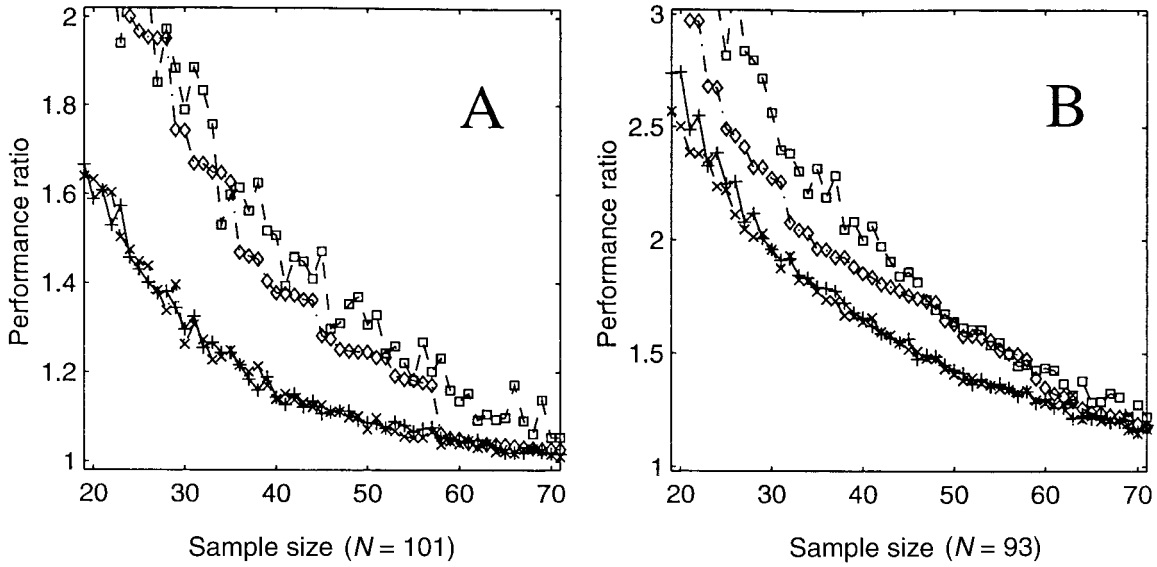

Figure 5.—Analysis of existing mapping populations. Shown are the performance ratios (of maximum bin length) for samples of varying size chosen by four different algorithms: (×) random-mixed greedy (minimizing maximum bin length); (+) randomized rounding (minimizing maximum bin length); (◇) most breakpoints; and (□) best of 50 random samples. (A) The Lister and Dean Arabidopsis recombinant inbred (RI) population. Population size is 101 and map length is 513.1 cM. (B) The Genebridge 4 human radiation hybrid (RH) panel. Population size is 93 and map length is 10,866 cR.

sizes were required for a given performance ratio and the sizes needed to approach a performance ratio of 1.0 were similar for all of the algorithms, including the naive ones. The explanation for this appears to be as follows. Consider a long genome composed of some number of short genomes concatenated end to end. The breakpoint distributions in the short genomes are independent and the MBL of the long genome is the maximum of the MBL among the short genomes. For any sample, the performance ratio for the long genome will be the worst performance ratio achieved for any of the short genomes. This will clearly be worse than the average performance ratio among the short genomes. Hence, the gain to be realized by selective sampling for MBL is diminished by increasing breakpoint density.

Table 1 shows the results of selective mapping for maximum bin size using 10 data sets, including the two described above. Samples were chosen, using several different algorithms to be approximately 30% the size of the base population. The randomized rounding and randomized mixed greedy algorithms performed comparably to one another and outperformed the most breakpoints sample in all populations. For the randomized greedy algorithm, performance ratios ranged from 1.29 to 2.2. While maximum bin length was significantly correlated with genome length (for the randomized greedy algorithm, $\rho = 0.8$, $P < 0.005$), performance ratio was not.

DISCUSSION

We have shown that, given genotypic data for a limited number of markers in a large mapping population, a much smaller sample can be selected for subsequent mapping that very nearly minimizes the necessary sacrifice in map resolution. Two computationally efficient algorithms have been found that are successful in finding samples with small maximum bin length. The first is a greedy algorithm that employs two objective functions sequentially, first minimizing the sums of the squares of the bin lengths and then minimizing the maximum bin length. It also exploits randomness to search a large space of possible good samples. The second algorithm involves linear programming with randomized rounding to convert fractional assignments into integral ones. Sample sets from both algorithms are improved by the inclusion of a clean-up routine that disposes of members that do not contribute to the quality of the sample and greedily replaces them with other selections that do. The greedy and linear programming algorithms dramatically outperform more naive alternatives such as choosing the individuals with the most visible breakpoints or choosing the best of a collection of randomly generated samples. Furthermore, the linear programming and greedy algorithms find samples that are within a few percentage points of the optimal sample for a given size, as indicated by comparisons with the integer programming solution for simulated data. For an alternative objective function, the sum of the squares of the bin lengths, the samples we find by using the randomized greedy algorithm are also superior in map resolution to random samples.

The radically different origins of the breakpoints in the various recombinant populations examined here and the GeneBridge 4 human radiation hybrid panel suggest that selective mapping is appropriate for a wide

TABLE 1

Maximum bin size in samples from a variety of datasets

Organism	Population size	Sample size	Map length	Whole population	Random rounding	Random greedy	Most breaks	Random sample
Arabidopsis[a]	101	30	510	4.5	5.9	5.8	8.0	10.3
Barley[b]	150	46	1,100	4.8	8.7	8.8	10.5	13.7
Barley[c]	73	22	1,000	8.0	14.9	14.3	17.3	24
Human[d]	93	28	10,900	15.4	21.8	20.7	23.8	38.4
Maize[e]	89	28	2,000	6.3	10.3	10.6	13.4	15.7
Mouse[f]	94	28	1,400	5.9	11.8	11.4	15.1	18.3
Mouse[f]	94	28	1,300	5.5	12.6	12.1	13.8	18.8
Rice[g]	98	28	1,200	4.8	9.5	9.3	12.8	13.6
Tomato[h]	67	19	1,300	6.7	10.4	10.8	12.9	16.4
Zebrafish[i]	96	28	3,100	11.3	17.7	17.8	19.2	26.6

All distances are measured in centimorgans except for the human radiation hybrid panel, which is measured in centirays.

[a] Lister and Dean (1993).
[b] Kleinhofs *et al.* (1993).
[c] Graner *et al.* (1994).
[d] Gyapay *et al.* (1996).
[e] Burr and Burr (1991).
[f] Blake *et al.* (1999).
[g] Taguchi-Shiobara *et al.* (1997).
[h] Tanksley *et al.* (1992).
[i] Postlethwaite *et al.* (1998).

variety of mapping populations. We have not analyzed multigenerational pedigrees here, because of the added complication due to breakpoints that are identical by descent. But these could, in principle, be accommodated. Our methodology does not require that any mapping function be used to derive the framework map. However, the assumptions made about breakpoint distribution in the sample selection process are the same as those underlying the Haldane mapping function (Haldane 1919). Under these assumptions, our methodology can accommodate segregating populations with unknown linkage phase (*e.g.*, an F_2) without further modification. If the model is violated by the presence of interference, then fewer double recombinations would be expected to occur between framework markers. As a consequence, the performance of the selection algorithms may be somewhat improved.

On the basis of these results, we propose a two-phase mapping strategy for projects in which a very large number of markers are to be mapped on a single population. In the first phase, the genotypes at a set of framework markers are scored in the full mapping population, and a framework map is constructed from these data. In the second phase, markers are mapped using only a selected subset of the population and positions are inferred relative to the fixed framework map. By adopting this strategy, a near-optimal balance may be reached between mapping precision and genotyping effort.

Practical considerations: The cost of genotyping is dependent upon laboratory methodology (Cotton 1996), but generally involves both a fixed cost per locus and an incremental cost per genotype. With selective mapping, the latter costs are reduced in direct proportion to the reduction in the size of the mapping population. For inherently serial genotyping methodologies, such as denaturing high-pressure liquid chromatography (Underhill *et al.* 1997) or mass spectrometry (Griffin *et al.* 1997), there would also be a directly proportional reduction in the time required to map each locus. Since mapping on the order of 5000 markers currently requires tens to hundreds of thousands of dollars and may take several years for a moderately equipped lab to complete, the potential savings from the application of selective mapping are considerable. Rather than sacrificing map resolution, one could apply selective mapping to a project with the aim of attaining the finest map resolution possible given the resources that are available by tuning the population size, sample size, and framework density accordingly.

Clearly, it is desirable to have a framework map sufficiently dense that one may catalog the locations of the overwhelming majority of the breakpoints in the population. But it would be counterproductive to make the framework so dense that only a small proportion of the markers remain to be genotyped in the selected sample. We propose the rule of thumb that, at the very least, framework markers should be chosen so as to be evenly spaced at intervals of less than half the length that would be tolerated as the maximum bin length. The rationale for this is that if two adjacent framework intervals each contain one breakpoint, the distance between the two outer markers is an upper bound on the length of the

TABLE 2

Effect of framework marker spacing on map resolution of selected samples

Spacing (cM)	MBL		SSBL	
	Perceived	Actual	Perceived	Actual
1	7.8 (0.7)	7.9 (0.9)	3.9 (0.2)	3.9 (0.3)
2	8.1 (0.6)	8.1 (0.7)	4.0 (0.2)	4.0 (0.2)
3	8.6 (0.5)	8.6 (0.3)	4.0 (0.2)	4.0 (0.2)
4	9.5 (0.4)	10.0 (1.1)	4.1 (0.2)	4.1 (0.2)
5	9.2 (0.3)	9.7 (2.0)	4.1 (0.1)	4.2 (0.2)
6	9.9 (0.2)	10.1 (0.7)	4.3 (0.1)	4.3 (0.2)
8	11.1 (0.2)	13.0 (1.5)	4.4 (0.2)	4.3 (0.2)
10	11.9 (0.3)	12.4 (0.8)	4.5 (0.1)	4.5 (0.2)
12	12.5 (0.3)	12.5 (1.8)	4.6 (0.1)	4.5 (0.3)
15	13.0 (0.6)	14.4 (3.0)	4.7 (0.2)	4.7 (0.2)
20	14.0 (0.7)	14.1 (0.9)	5.0 (0.2)	4.8 (0.4)
24	14.4 (0.8)	14.1 (2.5)	5.2 (0.2)	4.8 (0.3)
30	15.6 (0.4)	15.2 (3.1)	5.5 (0.2)	4.8 (0.3)

Mean and standard error (in parentheses) for samples of 30 from five simulated haploid F_1 populations of size 100 with genomes 600 cM long. Data for SSBL are divided by genome length and are thus expected bin size.

longest bin between them. If markers are spaced more widely, then the algorithms described in this article may perform poorly in finding a sample with a desirable breakpoint distribution. Table 2 shows the results of simulations in which marker density was varied on populations with exactly specified breakpoints with simulated genome length 600 cM, where samples were selected on the basis of the visible marker genotypes. Measured by MBL, both actual and perceived resolution varies roughly 2-fold between the highest (1 cM) and lowest (30 cM) densities, while for SSBL, it varies about 1.5-fold for perceived resolution and only 1.25-fold for actual resolution. Thus SSBL is more robust to framework density, although it tends to overestimate the expected bin length at sparse densities.

Investigators should also consider the base population size and the intended sample size prior to undertaking a high-density mapping experiment. The expectation of the MBL for the base population can be calculated *a priori* from the formula $E(\text{MBL}) \approx L \times \ln(rn + z)/(rn + z)$, where L is the length of the genome in centimorgans or centirays, r is the expected number of breakpoints per individual based on the type of population under consideration (*e.g.*, $r = 2L/100$ for an RI population derived from an F_2 cross by recurrent selfing), n is the size of the base population, and z is the number of chromosomes (which can be treated as if due to the occurrence of $z - 1$ uniformly distributed breakpoints; Feller 1957).

A further issue in sample selection is the need to avoid a sample for which multiple bins cannot be distinguished by genotype. This situation would create ambiguity in marker placement. In principle, even a small number of individuals allows for an extremely large number of possible genotypic configurations. For a mapping population of n individuals, in which one can distinguish x genotypes at each locus, there are x^n possible genotypic configurations. For reasonable values of n and x, the number of possible configurations is far larger than can be expected to occur in an actual mapping population (*e.g.*, $2^{30} > 1 \times 10^9$). However, the actual number of bins in a population is limited by the rarity of recombinational or radiation-induced breakpoints within each individual and will always be orders of magnitude less than x^n. It is governed both by the size of the population and the type (*e.g.*, in an F_2 population of size n, the expected number of bins is $2n + 1$ for a 100-cM linkage group). We have found that multiple bins with identical or near-identical genotypic configurations only occur in very small mapping populations and are not a major concern in practice (D. Brown and T. Vision, unpublished results).

The problem of data analysis after the second round of genotyping is somewhat novel, in that all distance information is derived from the observed distances in the framework map and potentially large numbers of new markers, genotyped only in the selected sample, are to be placed in ordered bins. We have developed fast and robust methods, to be reported elsewhere, that are appropriate to this analysis (D. Brown and T. Vision, unpublished results). Conventional mapping software may also be modified to order new markers relative to the framework and to each other and to measure distances relative to the fixed framework.

A natural extension of selective mapping would be to divide a very large population (of size much greater than 100) into multiple samples, each containing a desirable distribution of breakpoints for a particular chromosome, or region of the genome. These samples may be used to finely map a locus of particular interest after it has already been localized to a region by use of a selected whole-genome sample (Fain *et al.* 1996). This extension would allow high-density mapping at fine resolution with only a moderate increase in experimental effort over the use of a single genome-wide mapping sample.

The authors discourage application of selective mapping by individual investigators to community mapping projects, such as the interspecific mouse backcross maps being coordinated by the Jackson Laboratory. Due to the importance of monitoring data quality from different laboratories and the difficulties inherent in merging partial data sets, such projects require that genotyping be done on a common set of individuals.

Software availability: While the greedy and linear programming algorithms are nearly identical in performance, the greedy algorithm has the advantage that it does not require specialized linear programming software to implement and can be easily adapted to the SSBL objective function. On the basis of prototype Mat-

lab code with which we performed the tests in this research, we have designed software that implements the randomized mixed greedy algorithm for sample selection and that computes the locations of markers that have been genotyped in a selected sample relative to a user-supplied framework map. This software, called MapPop, allows one to select samples from populations that contain on the order of 500 individuals or less within minutes on a modern desktop computer. It is available from http://genome.cornell.edu/software.html along with documentation on its installation and use.

The first generation of saturated genetic maps (O'Brien 1990) has proven to be an invaluable resource for the identification and cloning of genes involved in crucial cellular functions and contributing to naturally occurring phenotypic variation in a wide variety of model organisms (Tanksley 1993; Collins 1995). These first generation maps are currently being greatly enriched by several new classes of molecular markers (Primrose 1998). While the first generation of saturated genetic maps included on the order of 1000 markers or less, the next generation of maps can potentially include tens of thousands of markers (*e.g.*, Wang *et al.* 1998). Such very-high-density maps will have a profound impact on efforts to characterize genomic structure and function, to understand the relationship between genotype and phenotype, and to compare these findings across distantly related taxa. If the effort necessary to produce very-high-density maps could be reduced by selective mapping, it would not only facilitate ongoing projects to generate such maps in model organisms but also allow these useful resources to be developed for other organisms.

D.G.B. acknowledges N. Edwards for his help in connecting Matlab models with CPLEX optimization software. The authors thank L. Rowe and M. Barter at the Jackson Laboratory for providing mouse data and K. Livingstone, D. Schneider, M. Sorrells, and S. Chasalow for helpful comments on the manuscript. T.J.V., D.G.B., and S.D.T. are supported by National Science Foundation (NSF) Grant DBI-98-72617, D.G.B by an NSF Graduate Research Fellowship and by the UPS Foundation, D.B.S. and D.G.B. by NSF Grants CCR-970029, DMS-9805602, and Office of Naval Research Grant N0014-96-1-00500, and R.T.D. by NSF Grant DMS-96-26201.

LITERATURE CITED

Ben-Dor, A., and B. Chor, 1997 On constructing radiation hybrid maps. J. Comput. Biol. **4:** 517–533.

Blake, J. A., J. E. Richardson, M. T. Davisson, J. T. Eppig and Mouse Genome Database Group, 1999 The Mouse Genome Database (MGD): genetic and genomic information about the laboratory mouse. Nucleic Acids Res. **27:** 95–98.

Burr, B., and F. A. Burr, 1991 Recombinant inbreds for molecular mapping in maize. Trends Genet. **7:** 55–60.

Chvátal, V., 1983 *Linear Programming.* W. H. Freeman, New York.

Collins, F. S., 1995 Positional cloning moves from perditional to traditional. Nat. Genet. **9:** 347–350.

Cotton, R. G. H., 1996 *Mutation Detection*. Oxford University Press, Oxford.

Crippen, G. M., 1991 Prediction of protein folding from amino acid sequence over discrete conformation space. Biochemistry **3:** 4232–4237.

Darvasi, A., 1997 Interval-specific congenic strains (ISCS): an experimental design for mapping a QTL into a 1-centimorgan interval. Mamm. Genome **8:** 163–167.

Fain, P. R., E. N. Kort, C. Yousry, M. R. James and M. Litt, 1996 A high resolution CEPH crossover mapping panel and integrated map of chromosome 11. Hum. Mol. Genet. **5:** 1631–1636.

Feller, W., 1957 *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, New York.

Graner, A., E. Bauer, A. Kellerman, S. Kirchner, J. K. Muraya *et al.*, 1994 Progress of RFLP-map construction in winter barley. Barley Genet. Newslett. **23:** 53–59.

Griffin, T. J., W. Tang, and L. M. Smith, 1997 Genetic analysis by peptide nucleic acid affinity MALDI-TOF mass spectrometry. Nat. Biotech. **15:** 1368–1372.

Gyapay, G., K. Schmitt, C. Fizames, H. Jones, N. Vega-Czarny *et al.*, 1996 A radiation hybrid map of the human genome. Hum. Mol. Genet. **5:** 339–346.

Haldane, J. B. S., 1919 The combination of linkage values and the calculation of distances between the loci of linked factors. J. Genet. **8:** 299–309.

Haldane, J. B. S., and C. H. Waddington, 1931 Inbreeding and linkage. Genetics **16:** 357–374.

Karp, R. M., R. Stoughton and K. Y. Yeung, 1999 Algorithms for choosing differential gene expression experiments. Proceedings of the 3rd ACM RECOMB Conference, Lyon, France, pp. 208–217.

Keats, B. J., S. L. Sherman, N. E. Morton, E. B. Robson, K. H. Buetow *et al.*, 1991 Guideline for human linkage maps: an international system for human linkage maps. Genomics **9:** 557–560.

Kleinhofs, A., A. Kilian, M. A. Saghai Maroof, R. M. Biyashev, P. Hayes *et al.*, 1993 A molecular, isozyme and morphological map of the barley (*Hordeum vulgare*) genome. Theor. Appl. Genet. **86:** 705–712.

Lincoln, S., and E. Lander, 1992 Systematic detection of errors in genetic linkage data. Genomics **14:** 604–610.

Lister, C., and C. Dean, 1993 Recombinant inbred lines for mapping RFLP and phenotypic markers in *Arabidopsis thaliana.* Plant J. **4:** 745–750.

Liu, B. H., 1998 *Statistical Genomics.* CRC Press, Boca Raton, FL.

Nemhauser, G. L., and L. A. Wolsey, 1988 *Integer and Combinatorial Optimization*. John Wiley & Sons, New York.

O'Brien, S., 1990 *Genetic Maps, Ed. 5.* Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

Paterson, A. H., J. W. Deverna, B. Lanini and S. D. Tanksley, 1991 Fine mapping of quantitative trait loci using selected overlapping recombinant chromosomes, in an interspecies cross of tomato. Genetics **124:** 735–742.

Postlethwait, J. H., Y. L. Yan, M. A. Gates, S. Horne, A. Amores *et al.*, 1998 Vertebrate genome evolution and the zebrafish gene map. Nat. Genet. **18:** 345–349.

Primrose, S. B. 1998 *Principles of Genome Analysis.* Blackwell Science. Oxford.

Raghavan, P., and C. D. Thompson, 1987 Randomized rounding. Combinatorica **7:** 365–374.

Resende, M. G. C., 1998 Greedy randomized adaptive search procedures. AT&T Labs Research Technical Report 98.41.1.

Staden, R., 1979 A strategy of DNA sequencing employing computer programs. Nucleic Acids Res. **6:** 2601–2610.

Taguchi-Shiobara, F., S. Y. Lin, K. Tanno, T. Komatsuda, M. Yano *et al.*, 1997 Mapping quantitative trait loci associated with regeneration ability of seed callus in rice, *Oryza sativa.* Theor. Appl. Genet. **95:** 828–833.

Tanksley, S. D., 1993 Mapping polygenes. Ann. Rev. Genet. **27:** 205–233.

Tanksley, S. D., M. W. Ganal, J. Prince, M. Devicente, M. Bonierbale *et al.*, 1992 High density molecular linkage maps of the tomato and potato genomes. Genetics **132:** 1141–1160.

Underhill, P. A., L. Jin, A. A. Lin, S. Q. Mehdi, T. Jenkin *et al.*, 1997 Detection of numerous Y chromosome biallelic polymorphisms by denaturing high-performance liquid chromatography. Genome Res. **7:** 996–1005.

Wang, D. G., J. B. Fan, C. J. Siao, A. Berno, P. Young *et al.*, 1998 Large-scale identification, mapping and genotyping of single nucleotide polymorphisms in the human genome. Science **280:** 1077–1082.

Communicating editor: G. A. Churchill

APPENDIX A: A CLOSED-FORM FORMULA FOR THE EXPECTED SUM OF THE SQUARES OF THE BIN LENGTHS

This section provides a closed-form formula for the expected sum of the squares of the bin lengths, or $E(\text{SSBL})$. We assume that we are given the number of breakpoints in each framework interval, over the whole sample, and given the lengths of the framework intervals.

Let $M = (m_1, \ldots, m_n)$ be the real-valued vector of framework interval lengths, and let $C = (c_1, \ldots, c_n)$ be the vector containing the number of breakpoints in each interval. Let I_1 be the first interval, I_2 the second, and so on. For example, if $M = (14, 20)$ and $C = (2, 3)$, there are two intervals. I_1 has length 14 and two breakpoints, and I_2 has length 20 with three breakpoints. Finally, let $F(M, C)$ be the expected value that we seek under the hypothesis that the C_i breakpoints in interval I_i are independent and uniformly distributed across that interval. We assume that the first and last markers are also breakpoints.

Proposition 1. *If the genome consists of only one framework interval, of length $M = m$, $C = c$, then $F(M, C) = 2m^2/(c + 2)$.*

Proof. By induction on c. The base case is if $c = 0$; then the only bin is the framework interval of length m, so $F(m, c) = m^2 = 2m^2/(0 + 2)$.

For the inductive case, suppose $F(m, c') = 2m^2/(c' + 2)$ for all $c' < c_0$. Consider $F(m, c)$. We analyze this by conditioning on the random variable X, which is the position of the last breakpoint in the interval. Then we see that the other $c - 1$ breakpoints in the interval $[0, X]$ are distributed independently and uniformly, so the expectation of the sum of the squares of their induced bins is $F(X, c - 1) = 2X^2/(c + 1)$ by our inductive hypothesis. The cumulative distribution function of X is $P(X \leq x) = (x/m)^c$, since if $X \leq x$, then all of the c uniform breakpoints occurred before x; each of these c independent events has probability x/m. Its density function is $p(x) = cx^{c-1}/m^c$. By the definition of expectation, then, we see that

$$F(m, c) = \int_0^m [(m - x)^2 + F(x, c - 1)]p(x)dx$$

$$= \frac{c}{m^c}\int_0^m [(m - x)^2 + \frac{2x^2}{c + 1}]x^{c-1}dx.$$

This reduces to $m^2\,((2c + 2)/(c + 1)(c + 2)) = 2m^2/(c + 2)$, as desired.

To extend this to the case of multiple framework intervals, we first consider the two-interval case, where $M = (m_1, m_2)$ and $C = (c_1, c_2)$.

Proposition 2. *In this case, $F(M, C) = F(m_1, c_1) + F(m_2, c_2) + 2m_1m_2/[(c_1 + 1)(c_2 + 1)]$.*

Proof. The case when either $c_1 = 0$ or $c_2 = 0$ is straightforward, since we add the length of the empty framework interval to the length of the first (respectively, last) bin in the other framework interval; for brevity, we omit it and assume both intervals have a nonzero number of breakpoints. Let X be the position of the last breakpoint in the first interval and Y be the position of the first breakpoint relative to the beginning of the second interval. These variables are independent in our probabilistic model. We expect that it is the bin between these two breakpoints that may make the calculations difficult. By analogy with the proof of Proposition 1, we see that

$$\begin{aligned} F(M, C) &= \int_0^{m_1}\int_0^{m_2}[F(x, c_1 - 1) + F(m_2 - y, c_2 - 1) \\ &\qquad + ((m_1 - x) + y)^2]p(y)p(x)dydx \\ &= \int_0^{m_1}\int_0^{m_2}\left[\left(\frac{2x^2}{c_1 + 1} + (m_1 - x)^2\right) + \left(\frac{2(m_2 - y)^2}{c_2 + 1} + y\right)^2 \right. \\ &\qquad \left. + 2y(m_1 - x)\right]p(y)p(x)dydx \\ &= F(m_1, c_1) + F(m_2, c_2) + \int_0^{m_1}\int_0^{m_2} 2y(m_1 - x)p(y)p(x)dydx. \end{aligned}$$

The last equality follows because each of the first two groups of terms depends only on one of the variables; when separated, these integrals are exactly analogous to those from the proof for Proposition 1. After some simple calculus on the remaining integral, given that the distribution of X has already been found and the distribution of Y is easily seen to be similar, we find that $F(M, C) = F(m_1, c_1) + F(m_2, c_2) + 2m_1m_2/[(c_1 + 1)(c_2 + 1)]$, as claimed.

A simple extension and symmetry argument shows that, under the assumption that all framework intervals (except possibly the first and last) contain breakpoints,

$$F(M, C) = \sum_{i=1\ldots n} F(m_2, c_i) + \sum_{i=1\ldots n-1} \frac{2m_im_{i+1}}{(c_i + 1)(c_{i+1} + 1)}.$$

For framework intervals with no breakpoints, the calculations become a bit more tedious. Here, if $c_i = 0$, there is a bin from the last breakpoint of I_{i-1}, to the first breakpoint of I_{i+1}; the square of its length must be properly added to the sum. Further computation shows that we must only add the term $2m_{i-1}m_{i+1}/[(c_{i-1} + 1)(c_{i+1} + 1)]$, which controls for the interaction between intervals I_{i-1} and I_{i+1}. This term is exactly analogous to the one that computed the additional contribution of the bin from x to y in the proof of proposition 2. We must also still include the interaction terms between intervals I_{i-1} and I_i and between I_i and I_{i+1}, as before.

We can assume that there are never consecutive framework intervals without breakpoints. Such intervals could be combined into a larger single interval with no

breakpoints, surrounded on both sides by nonempty intervals. Hence, no further cases need be considered. The formula for $F(M, C)$ is

$$F(M, C) = \sum_{i=1\ldots n} \frac{2m_i^2}{c_i + 2} + \sum_{i=1\ldots n-1} \frac{2m_i m_{i+1}}{(c_i + 1)(c_{i+1} + 1)} + \sum_{i:1<i<n, c_i=0} \frac{2m_{i-1}m_{i+1}}{(c_{i-1} + 1)(c_{i+1} + 1)}.$$

APPENDIX B: CHANGES TO THE LINEAR PROGRAMMING MODEL FOR THE CASE OF INEXACTLY SPECIFIED BREAKPOINTS

This section considers the somewhat different linear program used for the experiments with data from real populations. Here, as in appendix a, we do not know the exact site of the breakpoints, but only the number of breakpoints in a particular framework interval, so our earlier linear program, which specified exactly how far each breakpoint was from its next neighbor, is no longer appropriate.

We again consider a threshold approach. We want to model this new problem with a family of linear programs, where each linear program LP'_B is indexed by a threshold value B. As before, we seek a modeling strategy where LP'_B is feasible when there exists a sample S with expected maximum bin length not much larger than B.

Again, we have decision variables y_i for every population member $i = 1, \ldots n$. Let I_j be the interval between the jth framework marker and the $(j + 1)$st marker, with length d_j. For each framework interval I_j, let R_j be the set of all population members that have a breakpoint in that interval. As before, we have a constraint $\Sigma_i\, y_i = k$, which ensures that we choose only k population members.

Consider a given framework interval I_j of distance d_j, and suppose $d_j > B$. Suppose we want to bound the expected length of the maximum bin inside I_j. A standard probability result shows that if there are c independent, uniformly distributed breakpoints in this interval, then the expected maximum length between them is very close to $d_j(\log c + \frac{1}{2})/c$ (Feller 1957). Hence, if we want the expected maximum distance between consecutive breakpoints in the framework interval to be less than B, we must find the minimum c such that $B > d_j(\log c + \frac{1}{2}/c)$ and attempt to place c breakpoints into that framework interval.

One can easily find the optimal c for a given framework interval length d_j and desired maximum length B by, for example, Newton's method. We have approximated c by $\lceil 5d_j/B \rceil - 3$, which is very close to the correct value, is very quick to compute, and has given good performance in our experiments. So our constraint for this interval is $\Sigma_{i \in R_j}\, y_i \geq \lceil 5d_j/B \rceil - 3$.

If the length of framework interval I_j is less than B, we simply join consecutive intervals together, starting with interval I_j, until we have a new interval of length B or longer. Call the resultant macrointerval M_j, with length d'_j; suppose it contains the intervals I_j through I'_j. In this macrointerval, we analogously require that there are $\lceil 5d'_j/B \rceil - 3$ breakpoints. The breakpoints in these short intervals provide endpoints for the bins that end in subsequent, possibly longer intervals, and the creation of these macrointervals allows us to incorporate these breakpoints into the linear program; otherwise, the threshold would only be relevant for longer intervals.

Our resultant linear program, for a given threshold B, is the following, which we call LP'_B:

$$\sum_i y_i = k; \tag{7}$$

$$\sum_{i \in R_j} y_i \geq \lceil 5d_j/B \rceil - 3, \quad \text{for each interval } I_j \text{ such that } d_j > B; \tag{8}$$

$$\sum_{k=j\ldots j'} \sum_{i \in R_k} y_i \geq \lceil 5d'_j/B \rceil - 3, \quad \text{for each macrointerval } M_j; \tag{9}$$

$$0 \leq y_i \leq 1, \quad \text{for all } i = 1 \ldots n. \tag{10}$$

As before, there is a threshold B^* for which LP_{B^*} is feasible, but LP_B is infeasible for any $B < B^*$; with a very small amount of computation, we can compute the value of B^* to a high degree of accuracy.

However, a feasible solution to these constraints, even where all y_i are 0 or 1, does not guarantee that there exists a sample with expected objective function value less than B. For example, while two consecutive intervals may each have expected maximum bin length less than B, the global expectation may be greater than B. Still, feasible assignments to the decision variables do ultimately perform well with randomized rounding.