A computationally novel way to place new markers onto genetic maps

(Extended Abstract)

Daniel G. Brown* Todd J. Vision†

September 30, 1999

Abstract

We study the problem of extending genetic linkage maps to include a large number of new markers. We note that this problem should be addressed by placing new markers into the breakpoint-induced bins of the mapping population, rather than by attempting to infer marker order and distance from recombination fractions among closely-placed markers, which are the approaches of current software. Based on this observation, we have constructed a new approach for the placement of new markers onto framework maps which is extremely fast and highly accurate, even when executed on small-sized mapping samples. Further, our approach provides an estimate of the error of the placements for new markers, so investigators will know how precise a new marker's placement is expected to be. Unlike existing methods, our methods scale well as the number of new markers increases. We have tested these methods using both simulations and real biological data, and verified that both the placement of new markers and the estimation of the errors are precise.

*snowman@cs.cornell.edu. Department of Computer Science, Cornell University, Ithaca, NY 14853. Research supported by an NSF Graduate Research Fellowship, NSF grants CCR-970029, DMS-9805602 and DBI-98-72617, ONR grant N0014-96-1-00500, and the UPS Foundation.

†tv23@cornell.edu. USDA-ARS Center for Bioinformatics and Comparative Genomics, Cornell University, Ithaca, NY 14853.

1 Introduction

Genetic linkage maps are a powerful means of organizing information about the genetics of complex organisms. They provide a scaffolding for assembly of the fragments of whole-genome sequencing projects [14], and allow the isolation of genetic factors related to traits of biological, medical and economic importance [2, 16]. In genetic linkage mapping, the relative orders and positions of genomic landmarks called *markers* are inferred by use of a sample of individuals called a *mapping population*. This work discusses two proposed changes in the design of these experiments. The first, an experimental change, is to do most experimentation on a well-selected sample of the mapping population. The second, an analytic change, is a new method for placing markers onto the map.

For most of this century, laboratory technology was sufficiently crude that n different mapping populations were used to estimate marker order and recombination frequencies for n different non or partially-overlapping sets of markers. However, recent decades have seen tremendous advances in the technology for detecting subtle differences between alleles [3]. This has allowed a shift toward large-scale, very expensive experiments in which hundreds to thousands of markers are genotyped in a single mapping population [19]. In model organisms and organisms of commercial importance, the first-generation of whole-genome maps are currently being greatly embellished by the inclusion of expressed genes as markers. Such maps have the potential to be far more marker-dense than their predecessors, since many organisms of interest, such as angiosperms and mammals, are thought to possess tens to hundreds of thousands of expressed genes.

In what follows, we suggest an alternative approach to mapping such large numbers of additional markers in the usual case where a high-confidence, but relatively sparse, linkage map already exists. Our sampling proposal allows researchers to either construct higher-quality maps with greater precision than before, to map more markers onto a map, or to save money and time while still obtaining a high-quality map. Our mapping methods are much faster than existing ones, and attempt to solve a much more appropriate problem in placing new markers. We have tested our ideas on simulated and real data, and the results support our proposals.

Structure of the paper Section 2 discusses current methods used for linkage mapping. Section 3 describes our proposed changes to this design, summarizing our work on choosing population samples from mapping populations, and discussing our new approach to the location of new markers. In section 4, we give details of this new marker location method. We give results from our experiments on simulated and real populations in section 5. In section 6, we offer our conclusions.

2 Current approaches to these experiments

Basic background A simple example of a mapping population is a collection of second generation (or F2) progeny derived by inter-mating the first generation (or F1) progeny from two inbred, genetically homogenous, parental lines (See Figure 1). Such F2 progeny possess two different copies of each chromosome. As a result of *recombination* in the gametes of the F1, each F2 chromosome is a mosaic of segments descended from the two parental lines. The mechanisms for recombination need not concern us, but these events are rare.

Linkage maps are measured in units called centiMorgans, or cM. In 1 cM, we expect to see a recombination in 1% of the gametes in a generation. To a first approximation, disjoint regions of the genome recombine independently, so the Haldane model[6] assumes recombinations are induced by a Poisson process, where one recombination is expected every 100 cM. In linkage mapping, we seek the position of new markers, measured in cM, on the genome.

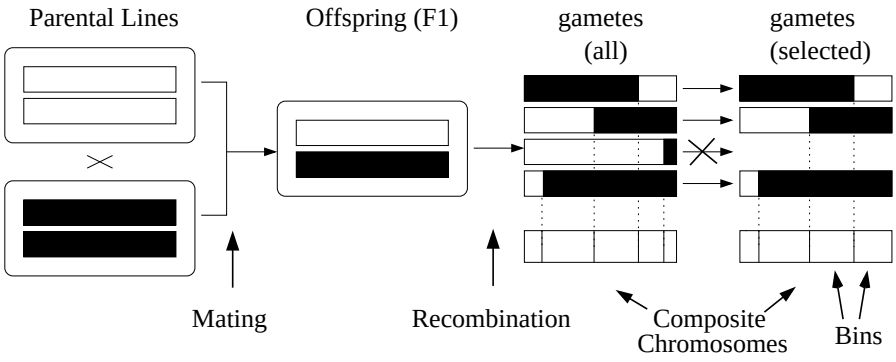

Figure 1: A mapping population derived from an F2 cross. Recombination in the gametes of the F1 results in mosaic chromosomes in the F2. Two "composite" chromosomes, one for an entire population and one for a sample, are shown. Each has a boundary at each position where a recombination event occurs in any member of the mapping population. *Bins* are the intervals between these boundaries, or *breakpoints*, and new markers cannot be placed more precisely than to the bin that contains them. A different sample of the population will result in a different composite chromosome and thus a different set of boundaries and bins. If all members contribute at least one breakpoint, removing of any member will necessarily result in fewer bins. Nonetheless, it is possible to choose good samples with good bin distribution.

The different copies of each marker in the two parental genomes are referred to as *alleles*. Assuming that there are differences between the alleles, various laboratory techniques are available that allow one to determine the *genotype*, or set of alleles, carried by each individual in the mapping population for a given marker. Since recombination is rare, if two markers are physically close on a chromosome, they will usually have the same genotype. Thus, relative orders and map distances among a set of markers can be inferred from the patterns of co-inheritance in a mapping population.

Current methods The methods for determining the order and position of markers on a large scale are derived from those used in small-scale experiments. First, markers are grouped into linkage groups. Then the markers in each linkage group are approximately ordered and the distance between adjacent markers is etimated. The first step is to assign markers to *linkage groups* from contiguous tracts of the genome. Ideally, each linkage group corresponds to one chromosome and each chromosome corresponds to only one linkage group. Assignment is performed by finding sets of markers, each of whose members show strong evidence of being linked with at least one other marker in the set. The most popular criterion for linkage is a log likelihood ratio known as *LOD score*[12]. This is derived from the likelihood that the two markers are linked relative to the likelihood that the two markers are unlinked, given the observed pairwise recombination fraction (RF), where RF is the proportion of chromosomes that do not carry alleles from the same parental line for both markers. In practice, RF values are first converted to additive, or nearly additive, map distances, which are used for all subsequent calculations. This grouping step is computationally rapid.

Subsequent to linkage group assignment, the goals are to obtain the optimal linear ordering of the markers for each linkage group and to estimate the map distances between all adjacent markers, given the order. Several objective functions have been used for the problem of finding an optimal order for a linkage group [10]. Two of the most commonly used algorithms combine ordering and distance estimation by either searching for an order that provides a good least squares fit to the matrix of pairwise map distances [7], or that that maximizes the likelihood of observing the mapping population, given the best fit map distances for a given order [13]. These approaches

are implemented in the popular software packages JoinMap [15] and MapMaker [9], respectively.

Most recent computational advances in linkage mapping have been improvements in the speed and efficiency of optimization over the space of possible orders and distances. This has been accomplished by means of algorithms such as simulated annealing [11] and expectation-maximization [4]. We note that similar problems are encountered and methods employed in the construction of so-called radiation-hybrid maps [18]. Although we focus on recombinational linkage maps here for ease of exposition, our methods are adaptable to other flavors of linkage maps.

An important technical feature of the current approach to placing new markers on a map is the continuous nature of the parameter being estimated: map distance between adjacent markers. A problem with this approach arises as maps become ever more densely populated with markers. As distance between adjacent markers becomes very small, the map positions of many markers become indistinguishable, since they localize to the same recombination-induced bin. Error in estimating the distance between adjacent markers, relative to the actual distance, becomes greatly inflated.

Another problem is that the multi-locus likelihood function and the goodness-of-fit test lack high power to discriminate between permutations of the ordering of very close markers. The likelihood curve is fairly flat for low map distances, so it is difficult to find a preferred order. Worse, the time required by these algorithms scales very poorly with the number of new markers to be mapped. Running time is linear for distance estimation given a fixed order[8], but finding the optimal marker order in practice very time consuming. Thus, while current methods are effective at constructing sparse whole-genome maps, they are ill-suited to mapping large numbers of new markers.

3 Our new proposals

We suggest two major changes to the structure of these high-density linkage mapping experiments. First, we recommend changes in the populations upon which markers are genotyped. In recent work [17, 1], we advise producing a framework map with a large, randomly chosen mapping population, using existing methods for joining markers into linkage groups and to order and locate the framework markers. Then, we recommend choosing a sample from the population after this framework map is produced, and performing further genotyping only on that sample of the population. Our previous work, briefly summarized below, discusses the optimization problems encountered in choosing a good sample upon which to continue experimentation. Our second structural change, which is largely the subject of this paper, is that new markers should be mapped into breakpoint-induced bins, and that finding the composite genotypes of these bins is more important than finding the order among new markers. This is briefly discussed below, and in detail in the following section.

Sample selection After producing a framework map, it is possible to approximately locate a large fraction of the breakpoints. For example, if we consider an F2 recombinant inbred population with framework markers every 10 cM, 90% of breakpoints can be identified because a population member has different genotypes at consecutive framework markers. Based on these perceived breakpoints, we can also approximately characterize the distribution of bin lengths in the population, or in any sample from it. Under the Haldane distance model, recombinations are uniformly distributed between their flanking framework markers. With this in mind, we can compute the expected distribution of bin length for a subset of the population. We expect that the usefulness of a sample for mapping should be a function of its bin lengths and use the inferred breakpoints to choose a good mapping sample, upon which we propose genotyping many more markers.

In our earlier papers, we considered three possible optimization approaches: maximizing the number of bins, minimizing the length of the longest bin, and minimizing the sum of the squares of the bin lengths. This last function may seem odd, but is equivalent to minimizing the expectation

of the length of the bin containing a marker chosen uniformly from the entire genome. If, for example, there are bins of length 3, 5 and 2 units, then the length of the bin containing a uniformly chosen marker is $\frac{3}{10} \times 3 + \frac{5}{10} \times 5 + \frac{2}{10} \times 2 = \frac{1}{10}(3^2 + 5^2 + 2^2) = 3.8$ units.

Optimizing the first of these is trivial–we simply choose the sample with the most visible breakpoints. But it may also be undesirable, since there may be many long regions of the genome with no breakpoints at all in them. The other two objective functions are harder to optimize (indeed, even if we know the exact sites of all breakpoints, they are still NP-hard to approximately optimize to within any constant factor). We have designed good heuristics for them using integer programming, linear programming with randomized rounding, and simpler randomized greedy methods. For a variety of different experimental populations, we have found population samples with significantly better map resolution, as measured by both expected bin length (sum of squares of bin lengths) and by maximum bin length, than the samples containing the most visible breakpoints or random samples. We conjecture that the most meaningful of these objective functions for mapping is the last, that of expected bin length, since it takes into account data from the entire genome. We have thus optimized this function when computing new samples for this study.

Mapping new markers with selected samples Several difficulties would present themselves if an investigator used existing mapping software to try to map thousands of new markers, genotyped on a selected sample, onto an existing framework map. First, the software cannot handle the computational overhead of trying to compute marker placements for so many marker orders, which yields unacceptable running times. Second, existing algorithms for building linkage maps infer genetic distance based on the RF of the population members. If we restricted the population to only the selected (and non-random) samples, this will result in significant distortion of the distances. These programs are not ideally suited to this scenario, though one could determine recombination fractions as a fraction of the known recombinations, to scale the mapping results.

A more serious objection, though, is that they do not attempt to solve the correct problem. The proper goal of high-density linkage mapping should be to assign markers to one of the finite number of bins induced by a given mapping population. No further resolution is possible, and estimates of low RF are not practically useful. A different method is needed.

We have designed a new approach to locate new markers, which first tries to find the most likely framework markers between which a new marker falls, then determines the composite genotype of the bins induced by the mapping sample, and finally assigns markers to the bins which match them. Our methods attempt to correct genotyping errors by looking for patterns which are not likely to be found in correct data. For simplicity, in the following description, we assume that the experimental population is a doubled haploid, with 50% of the genome of a population member expected to be derived from one of the lines' parents, and 50% from the other. We note that our methods extend to other types of populations (such as F2's), and can also be used with non-codominant markers, in a largely straightforward extension, again under the Haldane model.

4 Details of the mapping algorithm

Notation and problem specification First, we specify some notation. Let $F_{i,j}$ be the genotype of population member i on framework marker f_j, and let $N_{i,j}$ be the genotype of the population member i, on new marker n_j, and that all entries of these matrices are A, B, or ? (indicating that either A or B is possible, arising from an omitted genotype). For simplicity, let F_j be the entire sample genotype at framework marker f_j, and N_j be the same for new marker n_j. Let I_j be the genome interval between framework marker f_j and framework marker f_{j+1}. Let ϕ_j be the position of framework marker f_j on the genome. While this is unknown, we assume that its position was

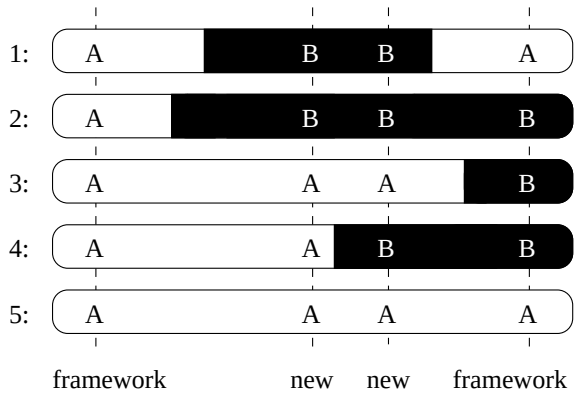

Figure 2: Assigning new markers to framework intervals and ordering bins. If a marker with a given genotype falls into a given framework interval, it may require unseen double breakpoint events. A new marker also constrains the order of breakpoint events between the framework markers. Here, the new genotype (B, B, A, A, A) requires a double breakpoint in chromosome 1, and that the first breakpoint in chromosome 1 and the breakpoint in chromosome 2 precede the breakpoints in chromosome 3 and 4. Otherwise, a bin with this genotype will not occur. The second new genotype of (B, B, A, B, A) lets us infer that the third breakpoint in this interval is in chromosome 4.

determined with a very large population, and is highly accurate. Let ν_j be the actual position of new marker n_j on the genome, which is unknown. We assume that genetic distance can be modeled by the Haldane distance model, where recombinations are generated by a Poisson process. Also, for simplicity, we assume that our framework markers include the telomeres of all chromosomes–that is, the entire genome is contained in the intervals between the framework markers; again, our methods can be applied when this is false. Finally, we assume that no framework intervals contain more than two breakpoints for any given population member. In real populations, triple recombinations are unlikely because of the well-documented phenomenon of interference [10].

Our problem is the following: given the framework genotype matrix F, the new marker genotype matrix N (and the possibility of noise in either of these matrices), and the framework distances ϕ_j, we are to identify the position of the bin containing each new marker n_j, and to estimate its position ν_j, based on the position of its bin. Further, if possible, we are to identify the composite genotype vector of as many of the genome's bins as possible. In our procedure, we first assign markers to framework intervals, then assign genotypes to bins, and finally assign markers to their correct bin. We also must cope with both noise and omissions in the genotype data.

Placing markers into framework intervals We first assign markers to the inter-framework intervals containing them. We do this on a marker-by-marker basis; in general, this process is quite accurate and very fast. Given a new marker n_j, for every inter-framework marker interval I_k, we compute the expected length $l_{j,k}$ of the region inside I_k which has genotype N_j, and scale this vector of lengths to a probability vector to determine the probability that the marker is in a given framework interval. Note that this is often infinitesimally small.

The process is as follows: First, we compute the number of hidden double breakpoints which need to have occurred for a genotype to be possible in interval I_k. These are lines i where both $F_{i,k}$ and $F_{i,k+1}$ differ from $N_{i,j}$, meaning that if marker n_j is in interval I_k, a breakpoint must occur in interval I_k on line i both before n_j's bin and after it. Next, of the lines i which have a visible breakpoint in I_k, since $F_{i,k} \neq F_{i,k+1}$, we determine those lines for which the breakpoint must occur after n_j (since $N_{i,j} = F_{i,k}$) and those for which the breakpoint occurs before n_j (since

$N_{i,j} = F_{i,k+1}$). This divides the population sample into four sets: those with no breakpoint, those with a required double breakpoint, those with a single breakpoint required before the new marker, and those with a single breakpoint required after the new marker. Suppose there are z, d, l, and r of these, respectively. This placement requires $l + r + 2d$ breakpoints in I_k (See Figure 2).

We can now approximate $l_{j,k}$. It is the expected length of a bin ($\frac{|I_k|}{l+r+2d+1}$) in the interval, times the probability that the required double breakpoints occurred (easily shown by reasoning about Poisson processes to be approximately $(\frac{1}{2}|I_k|^2)^d$ for small values of $|I_k|$), times the probability that, given these double breakpoints, the needed bin genotype actually occurs. If a bin with genotype N_j occurred, then all of the breaks in l, and one break from each line in d must occur before the breaks in r and the second break in each line of d. Assuming all breaks in I_k are independent, the probability of this combinatorial event is $2^d/\binom{2d+l+r}{d+l}$, giving this formula after simplification:

$$l_{j,k} = \frac{|I_k|^{2d+1}(d+l)!(d+r)!}{(2d+l+r+1)!}.$$

We scale these values to a probability vector. With this procedure, for any new marker genotype, we can infer its framework interval. The vast majority of new markers whose genotype is not the same as a framework marker are mapped to the correct interval with very high confidence ($p > .99$).

Determining bin genotypes If the number of new markers to place is small, this first step may be all that is possible. However, if the number of new markers is large (perhaps, at least as large as the number of bins in the genome), it is likely that we will be able to identify bin genotypes and place new markers directly into their correct bin.

Our goal in identifying bin genotypes is to order the breakpoints in each inter-framework interval; each new breakpoint induces a new bin. We examine the markers n_k that have been placed into each inter-framework interval I_j with probability $p_k > .5$. As noted above, each new marker n_k, placed into I_j, implies a constraint on the order of the breakpoints in the interval: a set U of breakpoints must precede another set V of the breakpoints in I_j. For each pair (u, v) in $U \times V$, we increase the votes that u precedes v by p_k. At the end of the procedure, we infer the order of each pair of breakpoints by which of the two orders has more votes, and construct an order of all of the breakpoints which is compatible with this pairwise order. From this ordering, we can infer the genotype vector of each bin contained within the interval and their relative order. As the number of markers correctly placed in I_j gets larger, the heuristic order obtained by this procedure will converge to the right bin order, even with randomly placed noise. We never experienced contradictory orders in our experiments.

To include lines with two breakpoints in I_j in this process, we must first order the bins showing the double breakpoint, and then place the bins on either side of them. The situation theoretically could become potentially complicated, but multiple double breakpoints are rare, and tend to be easy to order in practice. For simplicity, we discuss only the case where all lines have single breakpoints.

After this bin genotype assignment step, we determine the expected position of a marker uniformly placed in each bin, the approximate length of each bin, and the expectation of the square of the position of a marker placed in each bin. The expected length of any bin entirely enclosed in an interval I_j to which we have assigned b_j breakpoints is $|I_j|/(b_j + 1)$. The expected position of a marker in the kth such bin is $\phi_j + |I_j|(k + 1/2)/(b_j + 1)$, assuming the breakpoints are uniformly placed inside the interval and the new marker is uniformly placed in the bin. Given that the position of the left and right endpoints of a bin (under our Poisson assumption) inside their flanking interval are distrbuted as beta variables, we can also compute the expectation of the square of the position of a marker placed in each bin, which will allow us to estimate variance of marker placement. We

can find the expected length, expected position, and expected squared position of markers found in bins which include one or more framework markers as well.

Assigning new markers to bins With noise-free data, especially with samples of moderate size, the majority of markers will have a bin that matches their genotype (as was our experience in simulations). The other markers are either noisy, or an error has occurred in assigning genotypes to bins (or, possibly, multiple bins have the same genotype). For markers that exactly match only one bin genotype, we assign them to that bin with probability 1.

We assign the other markers to bins based on the expected lengths of the bins and the number of marker genotypes we must change to make a marker's genotype compatible with a bin's genotype. We assume that these switches, possibily caused by genotyping errors, occur with a user-specified probability p_e ($p_e = 0.01$ is a standard value). We assign a new marker to a bin with measure equal to the expected length of the bin, multiplied by p_e times the number of required changes, and then scale this measure to a probability vector which estimates the probability that a marker is in a bin.

In most cases, these bin placement probability values are very strongly focused on a single bin. In some very rare cases, however, two distant parts of the genome are both suggested to be possible. This only happened in our experiments (even under noisy conditions) for samples of size 20. Markers with two perturbed genotypes may be in bins whose genotype has been mis-identified (not surprising, since it may have had no other new marker in it). Such a marker may need as many as four or five changes to make it compatible with an inferred bin genotype for a place near its actual site on the genome. In the small space of $2^{20} \approx 10^6$ bin genotype vectors on 20-tuples, this perturbed new marker genotype may also be close to the genotype of a bin a far distance away on the genome. In our simulations, these failures were easy to identify; the marker was placed with probability greater than 0.1 in two distant bins. Should this happen in real experiments, we would suggest that the marker be re-genotyped. This serious error never occured in mapping samples of size 30 or higher. It would also be much less common in a denser framework, in which hidden double breakpoints are much less common and bin genotype inference is more precise.

Far more commonly, a marker is mapped either to one bin with very high probability, placed entirely into nearby bins. Here, we compute the expectation of the marker's placement, given the expected placement of a marker in each bin, computed in the previous step, and the expectation of the new marker's placement squared. These expectations can be used to estimate the variance in the new marker placement immediately, with no additional statistical procedures.

Noise and missing data Many complications to this approach arise when data may have noise or genotypes are unavailable for certain markers on certain population members. We have added several changes to this basic procedure to deal with these errors and omissions. First, if new marker genotypes are assumed to be in error with probability p_e, and a new marker has a genotype that disagrees with both of the flanking markers in a framework interval, this could be caused by either a double breakpoint, or by a genotyping error (or by the marker not actually belonging in that interval, of course). In most dense frameworks, genotyping error is much more common, so we raise the probability of seeing such an event accordingly. Similarly, we ignore population members which are omitted for a new marker when doing mapping, and assume the fewest needed double breakpoints when adding a new marker to a framework interval where one of the framework markers is untyped for a particular population member.

After we have assigned markers to framework intervals, we look for framework markers for which all of the markers assigned to the adjacent intervals on either side all have incompatible genotype for a particular population member; this suggests that either the framework marker is the only member of a double recombinant bin (which is possible, but unlikely), or that a genotyping error

has occured. We flag these framework genotype matrix entries as likely to be wrong, change them to omissions, and run the first step again.

To correct new genotype errors, we try to do much the same process: we look for markers which have been placed into a bin with high probability ($p > .9$), which are the only witnesses of a double breakpoint in their flanking interval, mark them as likely to be in error, and redetermine a placement for the marker. While single markers showing a double recombination are possibly correct, this can be validated by experimental re-genotyping, and by forcing the marker to be placed with unchanged genotype. Similarly, if a marker with missing genotype is placed into a framework interval in which both flanking markers and all new markers placed into the interval are of one genotype, we suggest that it is likely that the new marker is also of that genotype.

When we simulated a 1% genotyping error rate, these error-correction procedures were able to detect over half of the errors on mapping samples of size 20; in larger samples, they caught significantly more. We do not suspect, however, that these methods would work well for data with systematic errors and omissions, or with a high frequency of such difficulties.

5 Experimental results

The populations: real and simulated In our previous paper, we examined data from ten experimental populations, adapting existing linkage mapping data to our methods. To test the ideas in this paper, we need a large mapping population, typed on a large number of markers, with a small, scattered number of cells in the genotype matrix of unknown genotype. The one population from our original ten meeting all of our needs is a 73-member doubled haploid barley population, derived from a cross between IGRI and FRANKA, genotyped on over 470 markers [5]. We removed three population members genotyped on fewer than half of the markers.

Our other experiments were performed on ten simulated populations of size 100. We divided their 1000 cM genome into 200 cM and 800 cM linkage groups, to allow comparison with the largest barley linkage group, which is 200 cM long. We generated breakpoints via independent Poisson processes. Then, we simulated a framework map, with one framework marker placed every 15 cM and the genotypes of 500 new markers, placed uniformly at random in the 200 cM linkage group.

Experiments on simulated data In our experiments on simulated data, for each of the ten simulated populations, we computed eight samples of sizes $\{20, 30, \ldots, 90\}$, minimizing expected bin length, and eight samples of the sampe sizes maximizing the number of visible breakpoints. We computed ten random samples for each sample size. We mapped the 500 new markers onto the population for all of these 960 samples, and also for the ten full populations.

We compared the results from these 970 experiments, which took 4 hours on a Sun Ultra Sparc 20 workstation, against analysis of the same mapping populations using MapMaker 3.0, a standard genetic linkage mapping package[9]. We restricted analysis to only twenty of the new markers (in five clusters of four nearby markers), since MapMaker's running time would be far too long for the full set. In our tests, we used MapMaker's `place` and `together` commands, with the multipoint linkage criteria set to values which are very liberal in assigning markers to unique framework intervals. In these tests, we also used the Haldane mapping function.

Our experiments test four hypotheses. The first is that our algorithms would map the new markers more accurately as the sample size increased, but that even for well-chosen small samples, we would still be able to map with precision. The second is that MapMaker would have more difficulty with small samples, but that the two procedures would approach each other in quality as the sample size increased. The third is that the function which we minimize when picking samples, expected bin length, is closely related to what we really want to optimize, sample mapping

performance. The fourth is that the estimate of new marker placement error is accurate. That is, the distance from a marker's actual placement to the algorithm's placement is asymptotically normal, with mean 0 and standard deviation equal to our estimate.

Figure 3 shows the improved performance of our algorithms, as compared to MapMaker, as sample size increases, and the improved performance of our optimized samples, as compared to random samples or samples chosen to maximize the number of bins. The measure of performance is the root-mean-squared distance from the algorithm's placement of a marker to the actual placement. In MapMaker, we scaled the data so that the inter-framework intervals were the correct 15 cM, even when MapMaker shrank or expanded the region, to make a fair test. We do not include the markers which were not placed by the algorithms in these plots; this is much higher under MapMaker than under our methods. The optimized samples are much better for mapping than the random ones, and a Wilcoxon test on the rms error results for the most-recombinant samples and the optimized samples shows that the optimized samples perform better with high confidence ($p = 0.017$). The difference is moderate, especially for larger-sized samples.

Figure 4 shows that the performance of a sample for mapping is highly correlated with the measure which we attempt to optimize, expected bin length. We also note an odd difference between the three types of samples–the slope of the regression line is lower for random samples than for selected samples. That is, if we decrease expected bin length by 1 cM, mean absolute error goes down by 0.45 cM for the most recombinant samples, and by 0.46 cM for the optimized samples, but by only 0.38 cM for random samples. We do not know the source of this deviation.

Figure 5 shows the high quality of our error estimate. We took the positions estimated using the full population to be correct, and calculated the deviations from these positions for the same markers in the sample-derived maps. For an ideal error estimate, the ratio of deviation to estimated error would be normally distributed with mean zero and variance one. In fact, the ratio is distributed approximately normally, with mean 0.01 and standard deviation 1.10. The distribution has heavy tails, indicating that infrequently, the procedure underestimates the likely error.

We also performed experiments to see how well our algorithms handled noisy data, by flipping a random 1% of the genotype bits in each simulated marker population and running the 970 experiments again. Our procedures experienced a 5% increase in root-mean-squared error under this condition for small samples, and the error estimate was as accurate as before. A slightly elevated number of markers were not placed with high confidence under this procedure for samples of size 20. For samples of size 30 or higher, performance was comparable to the previous tests.

Experiments for the barley data In our barley experiments, we mapped the markers from one 200 cM linkage group of the barley data, using both our methods and MapMaker. We chose 12 of the markers, roughly spaced at 15 cM intervals, to serve as a framework, and mapped the remaining markers onto that framework. We note that while this experiment is not what we desired (mapping a huge number of well-typed markers onto a very dense framework), these results give a sense of the quality of our methods. We mapped all 98 markers in the linkage group with the bin mapping procedure, while with MapMaker, we picked 26 markers at different locations along the linkage group and ordered them. We mapped the markers using the data from the whole population of 70 plants, and from an optimized subset of 25 plants, using both mapping procedures.

Our procedures were much more successful at mapping than MapMaker on small populations. The running time of five seconds for our algorithms on these data is also noteworthy. The root mean squared distance a marker moved between a population of size 25 and the full population was 2.2 cM for our methods, and 7.4 cM for MapMaker. Our methods were again effective at estimating their error. While our error estimate is not normally distributed, it is still of high quality–the mean error is 0.19 times our estimate, and the standard deviation of this error is 0.85

times our estimate. Figure 6 shows the position of markers under the sample against their position under the full population, and the width of a 2-standard-deviation band around each new marker.

For MapMaker, to supply an error estimation, we used the nonparametric statistical method of jackknifing. To approximate the error of an estimate which depends on a sample S of size k, we compute the sample standard deviation among the k estimates obtained by the same procedure, performed on the sets $S - \{i\}$ for each i in S. The claim is that if removing any member from the set has minimal effect on the estimate, it is likely to be precise, while if removing members of the set has large impact, the estimate is likely to be imprecise. With jackknifing, we computed the approximate sample standard deviation for the placement of the markers, using MapMaker. For each new marker, we scaled the framework interval containing it to the same size as it was in the full population framework map, so that localized size changes of the perceived genome length were not responsible for mapping error. Even with this assistance, the jackknife standard deviation was very poor as a predictor of mapping error. While it is possible that other placement error estimation procedures would be more successful, our simple estimator, which is instantly computed, is of very high quality. A plot comparable to Figure 6 is shown in Figure 7; for each marker, we show two-standard deviation error bars around its placement under the 25-member sample.

While our results are preliminary, they show that mapping accuracy and error estimation are better under our much faster methods than under MapMaker, at least as implemented.

6 Conclusions

We have validated two proposed changes in approach to the problem of adding large numbers of new markers to framework genetic maps. The first of these, discussed in our recent papers [1, 17], is to perform most genotyping on a well-chosen sample of the mapping population, rather than the population as a whole. The sample is chosen after the production of a framework map. The second of these changes, the primary subject of this paper, is to view adding new markers to the existing maps as the problem of deterimining the genotypes of recombination-induced bins of the genome, and assigning new markers to these bins, rather than to continuously placed locations.

For the problem of locating new markers, we have developed new methods, based on Poisson process properties, to assign new markers to the correct framework interval, to assign genotypes to bins, and then to assign markers to the correct bin. With our methods, we can estimate the position of new markers, and have developed a method to estimate the error in new marker placement.

We have performed experimental tests to justify both of these changes. In our experiments on simulated populations, we showed that our methods are more precise, especially on small populations, than existing methods, like those found in MapMaker. We also showed that a selected sample which optimizes one of the functions, expected bin length, which we studied in our previous work, performs better for mapping than either random samples or samples chosen to maximize the number of breakpoints. This measure is highly correlated with the mapping precision of a sample. We found that our estimate of error in marker placement is reasonably accurate. Finally, in a test on a real doubled haploid barley population, we showed that mapping on a well-chosen sample is highly accurate, and that our mapping error estimates are reasonably good in this case as well.

Our methods are available in software we are developing, called MapPop. They are much faster than those found in other packages for genetic mapping. Yet this is not surprising–many of these sophisticated tools do a very good job at identifying new linkage groups, assigning new markers to nascent linkage groups, and ordering the markers in new linkage groups: the tasks which we describe as the production of a framework map. We propose that our methods, which explicitly place markers into breakpoint defined bins, are to be preferred for the final stage of a mapping project in which large numbers of new markers are to be added to a well-defined framework map.

Acknowledgments We would like to thank David Shmoys and Steven Tanksley, for useful discussions and for their support, and Sam Cartinhour, for help with programming in PERL.

References

[1] D. G. Brown, T. J. Vision, and S. D. Tanksley. Selective mapping: a discrete optimization approach to faster, cheaper genetic mapping experiments. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000. To appear.

[2] F. S. Collins. Positional cloning moves from the perditional to traditional. *Nature Genetics*, 9:347–350, 1995.

[3] R. G. H. Cotton. *Mutation Detection*. Oxford University Press, Oxford, UK, 1996.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Statist. Soc.*, 39B:1–38, 1977.

[5] A. Graner, E. Bauer, A. Kellermann, S. Kirchner, J. K. Muraya, et al. Progress of RFLP-map construction in winter barley. *Barley Genetics Newsletter*, 23:53–59, 1994.

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

[7] J. M. Lalouel. Linkage mapping from pairwise recombination data. *Heredity*, 38:61–77, 1977.

[8] E. S. Lander and P. Green. Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Sciences, USA*, 84:2363–2367, 1987.

[9] E. S. Lander, P. Green, J. Abrahamson, A. Barlow, M. J. Daly, et al. MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics*, 1:174–181, 1987.

[10] B. H. Liu. *Statistical Genomics*. CRC Press, Boca Raton, 1998.

[11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[12] N. E. Morton. Sequential tests for the detection of linkage. *American Journal of Human Genetics*, 11:1–16, 1955.

[13] J. Ott. *Analysis of human genetic linkage*. Johns Hopkins University Press, Baltimore, USA, 1985.

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

[15] P. Stam. Construction of integrated genetic maps by means of a new computer package: JoinMap. *The Plant Journal*, 3:739–744, 1993.

[16] S. D. Tanksley. Mapping polygenes. *Annual Review of Genetics*, 27:205–233, 1993.

[17] T. J. Vision, D. G. Brown, D. B. Shmoys, R. T. Durrett, and S. D. Tanksley. Selective mapping: a strategy for optimizing the construction of high-density linkage maps. *Genetics*, 1999. Submitted.

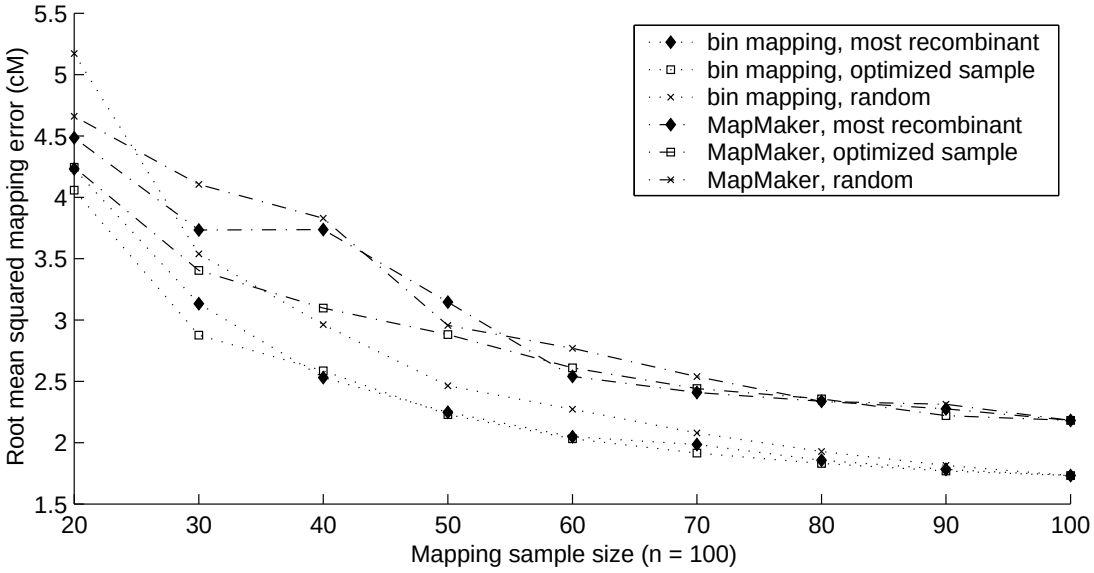

Figure 3: Mapping quality of simulated samples. This figure shows the root mean squared (rms) error of the mapping samples in our simulations. The bin mapping data is for 500 markers for each experiment, while the MapMaker data is for 5 clusters of 4 markers each. Optimized samples perform better for mapping, and our mapping algorithms perform better on smaller samples than does MapMaker; even for large populations, they are slightly preferred.

[18] M. A. Walter, D. J. Spillett, P. Thomas, J. Weissenbach, and P. N. Goodfellow. A method for constructing radiation hybrid maps of whole genomes. *Nature Genetics*, 7:22–28, 1994.

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

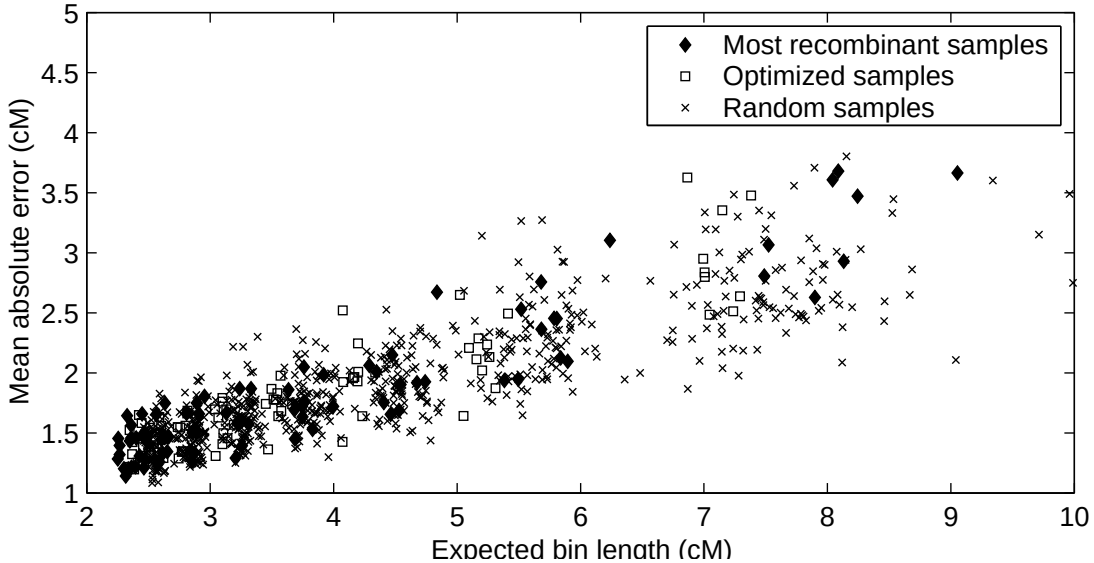

Figure 4: Scatter plot of sample mean absolute mapping error versus expected length. Data are from samples of various sizes on ten simulated populations of size 100 on a genome of length 1000 cM. For samples obtained either by choosing the most visibly recombinant population members, by minimizing expected bin length, or at random, mean absolute error is highly linearly correlated with expected bin length ($R^2 = .91$). The slopes of the regression lines of these three different sample selection methods are somewhat different.

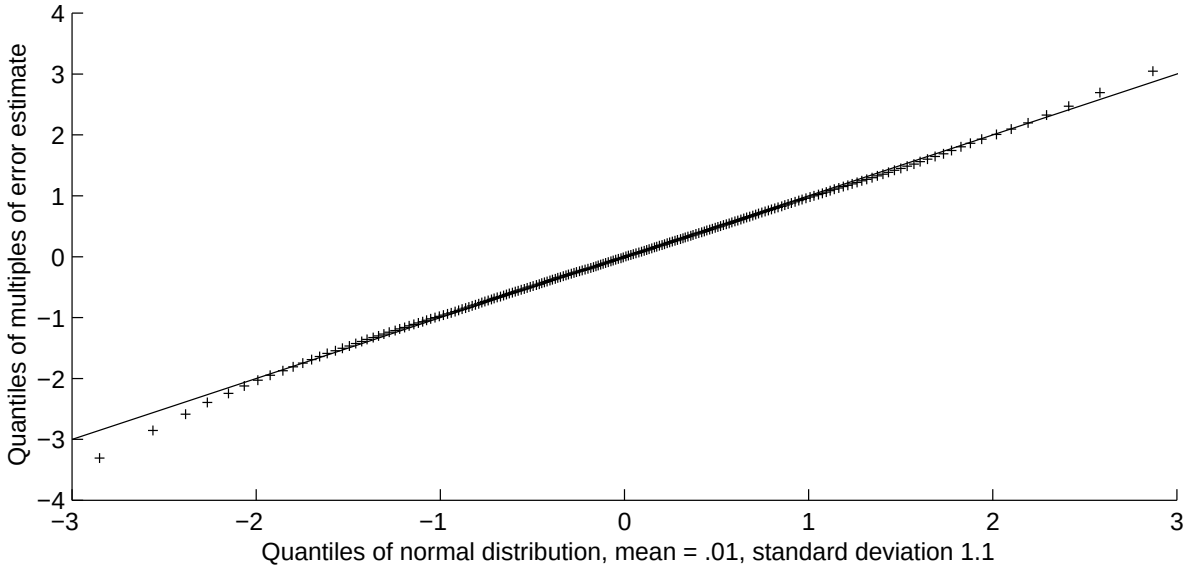

Figure 5: Quantile-quantile plot of simulation error distance, in multiples of the estimate of mapping standard deviation, versus quantiles from a normal distribution with mean 0.01 and standard deviation 1.1. If errors were normally distributed with these parameters, this quantile plot would be along the indicated line; instead, the simulated data has slightly heavy tails.

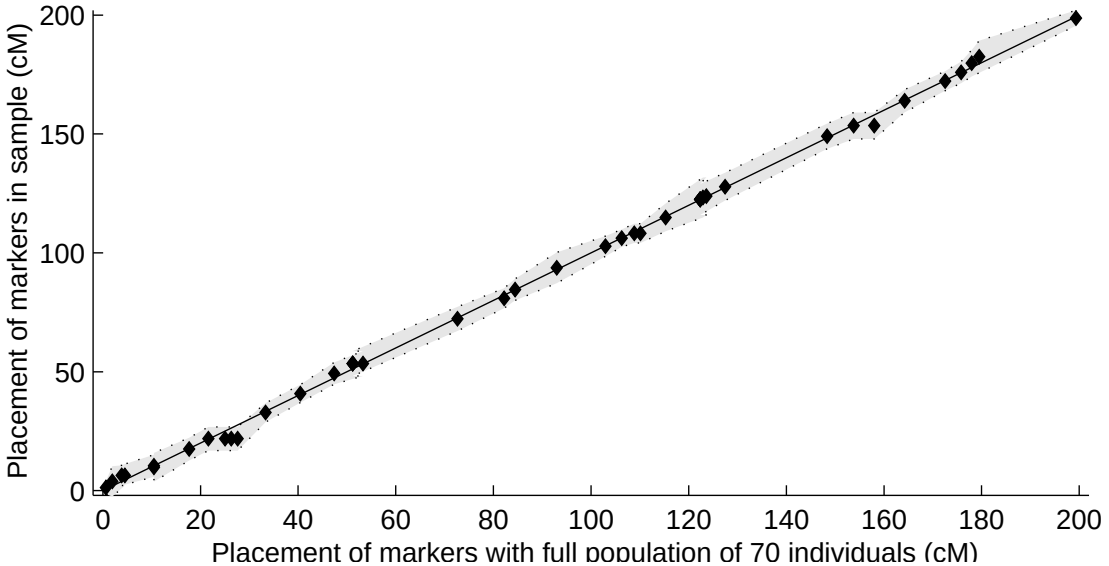

Figure 6: Placement of 98 markers on barley chromosome 7, from a doubled haploid population of 70 members. The expected position of the markers under a sample of size 25 is plotted against the position of the markers with the entire population. If mapping quality did not degrade for smaller sample sizes, the points would be along the indicated line. The shaded region is twice the predicted standard deviation in each marker placement; all but two of the markers are found in this region.

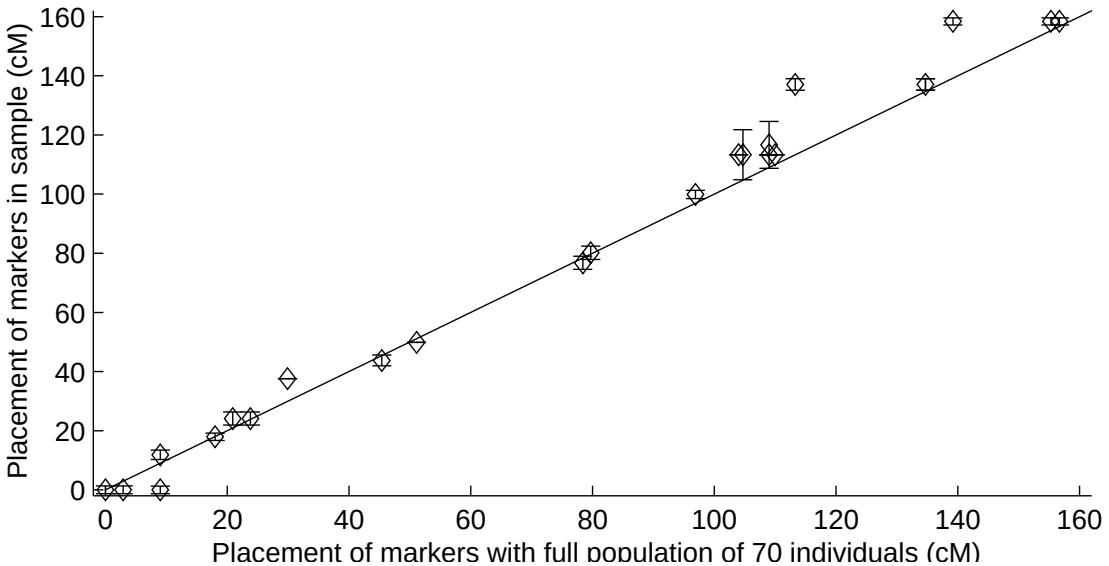

Figure 7: Placement of 26 markers on barley chromosome 7 under MapMaker. The axes are as in the previous figure, though the determined map length is somewhat different. Around each marker is an error bar of two times the jackknife error estimate in both directions. These estimates appear unrelated to marker placement error.