Determination of Marker Phases in Crosses with Many Offspring

Dustin A. Cartwright *†

June 3, 2007

Abstract

Motivation: Full likelihood analysis of large pedigrees with many markers is computationally intractable. In pedigrees which consist of a single pair of parents and their offspring, the complexity of the problem can be greatly reduced by fixing the marker phases in the parents. When there are sufficiently many offspring, their genotypes can be used to computationally determine the most likely parental phases. Traditionally, this step has been performed in any one of a number of *ad hoc* ways, which suffer from a lack of power and occasional mistakes.

Results: I present a systematic and efficient method for phase determination which works by maximizing the sum of the pairwise lod scores. This method compares favorably with the traditional pseudo-testcross method both on an actual data set in grapevine and on simulated data sets.

Availability: Phasing, the implementation used in this paper is available as part of the TMAP package from `http://math.berkeley.edu/~dustin/tmap/`.

1 Introduction

Full likelihood analysis of large pedigrees with many markers is computationally intractable, especially for mapping purposes, in which likelihoods of many possible orderings must be computed. Of the two algorithms available for likelihood analysis, the time required by the Elston-Stewart algorithm is exponential in the number of markers [4], and the Lander-Green algorithm is exponential in the number of non-founders [8]. Thus, with both many markers and many individuals, neither algorithm offers a viable solution. In genetic mapping, pedigrees frequently consist of two parents with a large number of offspring. One way to reduce the computational complexity of these pedigrees is to determine the phases of the markers in the parents. Because the progeny only relate to each other through their parents, gaining complete information about the parents means it is not necessary to analyze all the progeny jointly. In effect, each progeny, along with the parents, can be treated as a distinct pedigree, which can be analyzed efficiently by the Lander-Green algorithm.

An alternative approach is to avoid multipoint likelihood analysis altogether in large pedigrees. Two-point statistics, which look at only two markers at a time, can be computed efficiently for large, phase-unknown pedigrees. Furthermore, there exist algorithms for ordering markers based only on two-point statistics. For example, JoinMap uses two-point statistics to compute a map order, and thus, does not require phase-known data [12]. However, two-point statistics make less efficient use of the data than multipoint likelihood methods do, especially with missing data and mixed segregation types.

The phase of a heterozygous marker in an individual refers to which of the two alleles lies on which of the two homologous chromosomes. The chromosomes can be identified by whether they came from the individual's mother or from the father. However, when the individual's parents are not present in the pedigree, the chromosomes can only be identified by which alleles they have at another heterozygous marker. In these cases, all that matters is the

*Myriad Genetics, Salt Lake City, UT 84108

†IASMA Research Center, Genetics and Molecular Biology Department, 38010 San Michele all'Adige (TN), Italy

relative phase between pairs of markers, i.e. which allele from one marker lies on the same chromosome as which allele from the other marker. This paper only deals with the phases of the parents in a cross, and so the absolute phases do not matter. The expression *phase* without qualification refers to the phase in the parents for which the marker is heterozygous.

In a given pedigree, markers can be classified according to which parents the marker is heterozygous for. Markers which are heterozygous in both parents, called bridge markers, have independent phases in both parents for a total of four possible combinations, while markers which are heterozygous in only one parent only have two possible phases.

In some cases, the parental phases can be inferred from the genotypes of the grandparents and the rules of Mendelian inheritance. Backcross and intercross pedigrees are constructed so that the grandparents are always homozygous at every marker. Thus, for any marker heterozygous in a parent, the origin of the alleles is unambiguous. CEPH-type pedigrees also include genotypes from the grandparents. However, since the grandparents are not necessarily homozygous, only some of the allele origins are unambiguous [2]. Nevertheless, when the genotypes of the grandparents are completely unknown, or not sufficient to determine the phases, but there are many offspring, the genotypes of the offspring can determine the parental phases with very high probability.

The `chrompic` command in CRI-MAP finds the maximum likelihood parental phases from the genotypes of the offspring [7]. However, its method requires the marker order to be known, which makes `chrompic` more accurate, but it is not appropriate if the marker order is not already known.

In order to apply the `chrompic` command to a set of unordered markers, it is necessary to first determine a preliminary order using a method which does not require known phases [11]. While the exact marker order is not necessary to determine the phases, the maps must nonetheless be verified for plausibility. Thus, it is necessary to go through the work of producing the genetic map twice, first in preparation for finding the phases and second to produce the final map.

A second method for assigning phases to markers heterozygous in only one parent is to double each phase-unknown marker to produce two phase-known markers with opposite phases. In the doubled data set, each actual linkage group will result in two linkage groups, each copy consisting of markers with opposite phases. The copies of each marker which are in the same group are retained as the correct phases for those markers [6, 1, 10, 5, 9]. While this method has the advantage of not requiring any tools beyond finding linkage groups, this also means that the process requires a lot of manual recoding.

Furthermore, the marker doubling method does not extend well to markers in which both parents are heterozygous. Quadrupling these markers to all four possible phases does not work because a marker which is heterozygous in one parent will link to two copies of any nearby marker which is heterozygous in both parents. Thus, it is necessary to use a pseudo-testcross, meaning that the genotypes are projected onto each parent separately so that the phase in each parent can be determined independently. For each parent, the genotypes such that the identity of the corresponding gamete can be uniquely inferred are coded accordingly and as unknowns otherwise [6]. For example, in an $AB \times AB$ marker, progeny with genotype AA are coded as A for both parents, progeny with genotype BB as B for both parents and AB as unknown. Clearly, information about the AB genotypes is being thrown away by this method. The problem is even more pronounced in the case of dominant $A0 \times A0$ markers, where 0 denotes the null allele, in which three quarters of the progeny will have genotype A- and so be marked as unknown in the pseudo-testcross, meaning that it is very difficult to determine correct phases for these markers.

In theory, subsequent analyses which treat the parental phases as fixed to their most likely values suffer from biases. However, for large pedigrees, the probability of the most likely phases is so close to certainty that the magnitude of these biases is negligible.

Therefore, this paper presents an automatic and effective method of determining parental phases in crosses when the marker order is unknown.

2 Methods

Suppose we have a pedigree consisting of a cross between two individuals, referred to as the mother and the father and many of their offspring, all of which have been genotyped at many markers.

2.1 Phase determination with a single pair of markers

For a given pair of markers, the possible relative parental phases can be compared using their likelihoods. The phase-known likelihood of linkage between markers i and j, denoted $\text{Llhd}_{ij}(m, p, \theta)$, is a function of m, the relative maternal phase between markers i and j; p, the relative paternal phase; and θ, the recombination fraction. The lod value is defined to be the log likelihood ratio:

$$L_{ij}(m, p, \theta) = \log_{10} \frac{\text{Llhd}_{ij}(m, p, \theta)}{\text{Llhd}_{ij}(m, p, 0.5)}$$

Since we are interested in comparing the different relative phases and not in the recombination fraction, we maximize over the values of θ to obtain the lod score:

$$L_{ij}(m, p) = \max_{0 \le \theta \le 0.5} L_{ij}(m, p, \theta)$$

to get a function of just the relative phases.

If one or both of the markers is homozygous in one of the parents then the likelihood does not depend on the relative phase in that parent. In such cases, the genotypes of the offspring offer no evidence regarding the occurrence or absence of recombination between the two markers in that parent. For example, if marker i or marker j is homozygous in the mother, then

$$L_{ij}(m, p, \theta) = L_{ij}(\overline{m}, p, \theta)$$

where $\overline{m}$ denotes the opposite relative phase of m. A similar property holds when one or both markers are homozygous in the father. In these cases, only half of the lod scores need actually be computed.

Furthermore, in all cases, the lod score of a relative phase and the opposite relative phase can be computed at the same time. Usually, the recombination rate between two markers is limited to the range $0 \le \theta \le 0.5$, and values of $\theta > 0.5$ are considered meaningless. However, the values where $0.5 < \theta \le 1$ are equivalent to recombination fractions of $1 - \theta$, but with the opposite relative phase between the two markers from what was assumed, i.e.:

$$L_{ij}(\overline{m}, \overline{p}, \theta) = L_{ij}(m, p, 1 - \theta)$$

The likelihood function typically has a single maximum on $0 \le \theta \le 1$, denoted $\hat{\theta}$. If $\hat{\theta} < 0.5$, then maximum on the range $0.5 \le \theta \le 1$ occurs at $\theta = 0.5$, so the lod score of the opposite phase is zero:

$$\begin{aligned} L_{ij}(\overline{m}, \overline{p}) &= \max_{0 \le \theta \le 0.5} L_{ij}(\overline{m}, \overline{p}, \theta) \\ &= \max_{0.5 \le \theta \le 1} L_{ij}(m, p, \theta) \\ &= L_{ij}(m, p, 0.5) = 0 \end{aligned}$$

Similarly, if $\hat{\theta} > 0.5$, then $L_{ij}(m, p) = 0$ and $L_{ij}(\overline{m}, \overline{p}) = L_{ij}(m, p, \hat{\theta})$.

2.2 Phase determination with many markers

With many markers, it is no longer feasible to compute the likelihood of a given set of phases. Since the marker order is unknown, then the likelihood must include a summation over all possible orders, which is not computationally feasible. Thus, it is necessary to determine the phases of many markers from the two-marker likelihoods for all pairs of markers.

However, for three or more markers, maximizing the pairwise relative phases can lead to contradictions. For n markers, there are $n(n-1)/2$ pairs of markers. If each pair is to have its relative phase at the maximum pairwise likelihood, there are $n(n-1)/2$ constraints, but only $n-1$ relative phases, which leads to potential contradictions for $n \ge 2$. Ideally, all the pairwise relative phases would be consistent, but, due to noise or errors in the genotyping data, they might not be. It would still be possible to determine the phases based on a particular set of $n-1$ pairs of markers, but the solution would depend on the set chosen, and there is no obvious way to choose this set when the markers are unordered.

A more robust generalization to more than two markers is to choose the phases which maximize the sum of all pairwise lod scores:

$$\sum_{i<j} L_{ij}(m_i \oplus m_j, p_i \oplus p_j) \tag{1}$$

where $m_i \oplus m_j$ and $p_i \oplus p_j$ denote the relative phase between the two markers in the mother and the father respectively. Because the lod scores L_{ij} are, by definition, logs of likelihood ratios, addition is a natural way to combine them. If all the pairwise constraints defined above are consistent, then a solution to the pairwise constraints will also give the maximum sum to equation 1, because each term will be at its maximum. However, even if the pairwise constraints are inconsistent, equation 1 nevertheless has a maximum.

In order to more easily find a solution, we consider only the terms of equation 1 over a certain lod threshold T:

$$\sum_{(i,j)\in S_T} L_{ij}(m_i \oplus m_j, p_i \oplus p_j) \tag{2}$$

where

$$S_T = \big\{(i,j) | i < j, L_{ij}(m,p) > T \text{ for some } m,p\big\}$$

This eliminates most of the marker pairs which are not truly linked and whose corresponding terms in equation 1, therefore, do not improve the accuracy of the solution. The higher the value of T, the fewer terms equation 2, and thus, the easier it is to maximize, but also the less accurate equation 2 is as an approximation of equation 1.

As discussed above, whenever one of the markers in a term of equation 2 is homozygous in one of the parents, then the value of L_{ij} depends on only one of its parameters. In order to more efficiently find a maximal solution to equation 2, it is necessary to explicitly recognize this symmetry in the problem statement. Thus, some terms will be functions of single relative phases and some of two relative phases. In order to represent both cases using the same notation, but without excessive indexing, it is useful to introduce the following notation. We represent the phases as elements of the field of two elements, $\mathbb{F}_2$, which are equivalent to Boolean values where addition ($\oplus$) is defined as the exclusive-or operation and multiplication is defined as the and operation. Let n be the number of markers and let $\mathbf{p}$ be a column vector consisting of all $2n$ phases in both parents. Then, for each i, j, we can construct a $2 \times 2n$ matrix M with exactly two non-zero elements in each row such that

$$\begin{bmatrix} m_i \oplus m_j \\ p_i \oplus p_j \end{bmatrix} = M \cdot \mathbf{p}$$

The entries of this vector are the arguments of the function L_{ij}. Furthermore, if L_{ij} does not depend on one of the two relative phases, then the corresponding row of M can be dropped. Using this notation and reindexing, we can rewrite equation 2 as

$$\sum_k f_k(M_k \cdot \mathbf{p}) \tag{3}$$

where k indexes the elements of S_T and each M_k has either one or two rows, and each row has exactly two non-zero entries.

2.3 Maximizing equation 3

The basic structure of the algorithm is to perform a breadth-first search, building up partial solutions consisting of values for a subset of the phases until a single full solution is left. At each step, the partial solutions are uniquely determined by their values on a subset of the defined phases, which are called the index phases. The phases pass through the following three states in order:

1. Unknown, meaning the value is undefined in the partial solutions.
2. Index, as defined above.
3. Fixed, meaning the value is defined in the partial solution, but it is not a index phase.

The two fundamental operations are transitioning a phase from unknown to index, during which the number of partial solutions doubles, and transitioning a phase from index to fixed, during which half of the partial solutions are discarded, as described below.

Suppose we have two putative solutions which differ by only a single phase. Rather than compute equation 3 in its entirety, it is only necessary to compare those terms which depend on the differing phase. More importantly, it is not even necessary to know the values of those phases which never occur in any of the computed terms. Thus, given two partial solutions which differ by a single phase, and such that the defined phases include all phases which are used in all terms which depend on the differing phase, it is possible to eliminate one of them as non-optimal. Applying this to each pair of partial solutions which differ by only a single phase, in effect transitions the phase from index to fixed.

Furthermore, we can take advantage of the arbitrariness of the absolute phases. Define a graph whose nodes correspond to the phases and such that there is an arc between two nodes i and j if and only if there exists a row of some M_k which as entries of 1 at i and j. Reversing those phases which belong to any connected component of this graph will result in the same value for each $M_k \cdot \mathbf{p}$ and thus, the same value for equation 3. In other words, for each connected component, there is an arbitrary choice of phase. Therefore, whenever the first phase from each component is added to the set of index phases, we can immediately transition the phase to fixed by arbitrarily discarded one of each pair of partial solutions.

We can improve the efficiency by pruning the partial solutions more aggressively. In this case, we will not be able to compare the pairs of partial solutions on all of the terms of equation 3 in which they differ, so we compare them based only on those terms which do not depend on any unknown phases. Thus, partial solutions which are not provably sub-optimal are removed during the algorithm, so the final answer is not necessarily optimal. However, for each eliminated solution, we can compute the maximum possible final value of equation 3 by assuming the maximum possible value for each of the terms which depend on unknown phases. If the maximum possible values for all the eliminated partial solutions, E, is less than the the value for the final solution, then the latter is optimal, in spite of the short cuts. If not, then the algorithm fails, and must be repeated with a higher threshold T.

In order to minimize the chances of the algorithm failing, we want to minimize the value of E, the maximum of all the discarded solutions. At each step where we want to prematurely transition a phase to fixed, we can compute the resulting value of E assuming in turn, each of the possible phases to be transitioned, and choose the one which leaves E at a minimum. Furthermore, if any phase can be fixed without increasing E, then doing so saves time and memory without increasing the risk of failure.

This optimization is effective because the optimal solution is likely to have most of the terms of equation 3 also at their maximum values. In other words, the maximum solution is also likely to also have the maximum likelihood relative phases for most pairs of markers. Thus, for each prematurely fixed phase, the eliminated partial solutions, which assume optimality for the undefined terms of equation 3, will still have a lower sum than the optimal solution.

In my implementation, I put an upper limit on the number of index phases, currently 18. Whenever there are more index phases than this threshold, one phase is fixed. This value was chosen because it was the largest value which would not cause the operating system to swap excessively on the development machines.

Explicitly, the algorithm is given in figure 1. In order to make the notation cleaner, it processes the phases in the order in which the are in $\mathbf{p}$. As a consequence of this assumption, there is no need to keep track of the state of non-index phases; those less than the current value of i are fixed and those greater than the current value of i are unknown. As described below my implementation chooses the phase order based on the values of the f_k. The implementation of the algorithm in Figure 1 accepts the order as a parameter, which adds extra book-keeping to keep track of which phases are fixed and which are unknown, but does not change the fundamental algorithm.

While the optimality of the solution does not depend on the order in which the nodes are processed, the ability to find that solution does. My implementation tries each phase in each group of linked phases as a starting point. For each unordered phase, the following heuristics are computed:

Input: n, the number of phases; a list of matrices M_k of dimension $m_k \times n$, as described in the text; corresponding functions f_k given as tables of 2^{m_k} real numbers
Output: Either phases **p** such that equation 3 is maximized or failure

```
C ← 18
E ← −∞
array of (partial solution, partial sum) pairs ← {(all phases undefined, 0)}.
index phases ← {}
for i = 1 to n do
  if ith phase is the minimum phase of a connected component of the graph whose arcs are defined by the rows of the M_k then
    set ith phase to 0 in all partials
  else
    Duplicate each (partial solution, partial sum) and set the ith phase to the two possible values.
    Add i to the set of index phases
    for k such that the index of the rightmost non-zero column of M_k is i do
      Add f_k(M_k·partial solution) − max_r f_k(r) to value of each partial sum.
    D ← −∞
    for j in index phases do
      A ← −∞
      for pairs of partial solutions which differ only in jth phase do
        A ← max(A, min(pair of partial sums))
      if A > D then
        (ℓ, D) ← (j, A)
      if A < E or for all M_k such that the jth column is non-zero, all columns to the right of the ith column are non-zero then
        for pairs of partial solutions which differ only in the jth phase do
          Remove the partial with the lower partial sum
        Remove j from the index phases
    if number of index phases > C then
      for pairs of partial solutions which differ only in the ℓth phase do
        Remove the partial with the lower partial sum
      Remove ℓ from the index phases
      E ← D
(solution, sum) ← single element of partials array
if sum < E then
  return failure
else
  return solution
```

Figure 1: Explicit algorithm

1. The total number of non-zero columns which correspond to previously ordered phases in all the M_k for which the phase's column is also non-zero.

2. The sum of the magnitudes of the terms in equation 3 in which the phase is the only undefined variable. (The magnitude of a term is defined as its greatest possible value).

3. The sum of the magnitudes of all the terms in which the phase is present.

The phases are ranked in decreasing order according to the first heuristic, with ties broken by the the second, and ties in that broken by the third. The first phase according to this ranking is appended to the order. Finally, for each of these orders, the maximum number of index phases is computed, assuming no upper limit, and the starting phase and associated order are chosen. This last heuristic helps to pick orders which start at one end of the linkage group, not in the middle. Variations on all of these heuristics were tried and were less effective on the data set described below.

As a final step, the arbitrary choices of phases are reconsidered relative to each other. First, the relative phases within each component are fixed. Then, all pairs of markers are considered in order of decreasing lod score, $\max_{m,p} L_{ij}(m,p)$. Those pairs which are not already fixed are fixed according to the relative phases with the maximum lod score.

Finally, unless a value of T is explicitly given, the above algorithm is repeated to find the lowest value such that a solution can be found.

3 Results

The above algorithm was implemented in a C++ program called Phasing and tested on an actual data set of 1019 markers genotyped in a pedigree consisting of 94 offspring of a cross between the grapevine cultivars Syrah and Pinot noir (M. Troggio, unpublished data) and on simulated pedigrees. For comparison, the pseudo-testcross method, consisting of the following steps, was also used. First, each marker was projected onto one or both parents, as necessary, to make two separate data sets. Second, each marker was duplicated, one copy in each of the possible phases. Third, the resulting data sets were divided into linkage groups using the `build` command in CarthaGène version 0.999 [3] with a distance threshold of 40 cM Haldane and the lowest lod threshold which did not link markers to their own opposite phases. Finally, the phases which linked together were taken to be the correct relative phases and the original data set was recoded accordingly [6].

3.1 Grapevine pedigree

With Phasing, $T = 3.4$ was the lowest value for which the algorithm could find a solution on the grapevine data set. At this value, equation 3 consisted of 12,545 terms. The markers in the output were then divided into linkage groups using CarthaGène with a lod threshold of 7 and a distance threshold of 40 cM Haldane [3]. Markers in common with the International Grape Genome Project reference map [10] were used to verify that the data was correctly divided into 19 linkage groups corresponding to the 19 chromosomes of grapevine.

There are two classes of potential errors in the phase determination of a marker and they manifest themselves in different ways. Markers with half-correct phases (i.e. bridge markers with incorrect maternal phase and correct paternal phase or vice-versa) would still be linked because of nearby markers heterozygous only in the parent with the correct phase, but would have a recombination fraction of around one half because almost all of the progeny would appear to have recombinations in the incorrectly phased parent. Markers with fully incorrect phases would not even be linked with other groups, or would only be linked with other incorrectly placed markers.

To detect any markers which were unlinked because of with fully incorrect phases, all 21 unlinked markers were recoded with the opposite phases, and the resulting data set was again divided into linkage groups. However, the 21 recoded markers were still not linked with any of the other 998 markers.

To detect any markers with half-correct phases, and thus, large recombination fractions, maps were

built from each of the 19 groups using CarthaGène with the `build 5` command followed by `annealing` with 50 tries, an initial temperature of 50, a final temperature of 0.1, and a cooling factor of 0.9. There were seven pairs of adjacent markers which were separated by recombination fractions of more than a third, and each of these was examined to see if it was caused by a phasing error. All seven gaps were between one or two markers at the end of a linkage group and the rest of the markers in the linkage group. All possible alternative phases for these end markers were inserted into the data file, and linkage groups were found in the same way as above. Only three of the alternatively phased markers were linked with any of the nineteen linkage groups. New maps were made of the relevant linkage groups using the alternative phases for these three markers via the same method as above. Two of the resulting maps had lower likelihoods than with the original phases, validating the original phase choices. The remaining marker was given a less likely phase by Phasing.

Thus, out of the 998 linked and 21 unlinked markers, Phasing made only a single incorrect phase assignment.

Using the pseudo-testcross method on the full data set, the lowest possible lod threshold was 9.2, and the resulting phases were very inaccurate. By trial and error, two markers were found and removed so that it possible to use a lod threshold of 4.7. The loss of these two markers was more than offset by the increased accuracy afforded at a lower lod threshold.

Using the same method as above for finding linkage groups, there were only 993 linked markers. Five markers that were linked in the Phasing output were no longer in any linkage group. One of these was one of the markers which had to be removed from the data set, and another was only linked via the same removed marker. The other three unlinked markers were due to incorrect phases. In addition, there were six linked markers whose phases differed from the results from Phasing. In these six cases, maps were built with both phases using the same method as above. In five out of the six cases, the results from the pseudo-testcross had a lower likelihood than the Phasing results. The remaining case was the error in the Phasing output already identified above.

Threshold	Incorrect markers
4.7	9
5.0	10
5.5	11
6.0	35
6.5	38
7.0	42

Table 1: The accuracy of pseudo-testcross method on the grapevine data set using different lod thresholds.

In summary, if the pseudo-testcross method were used to determine the phases instead of Phasing, five markers would be lost because they would be unlinked, and a net of four linked markers would be lost due to incorrect phases.

3.2 Higher lod thresholds

Phasing was re-run with higher values of T, every 0.5 from 3.5 to 7.0. The results were equivalent to the original results at $T = 3.5$ and 4.0, but two additional markers had phase errors at $T = 4.5$. Over the entire range, only five markers had incorrect phases at any value of T, including the marker which was incorrectly phased at $T = 3.4$. The other four markers consisted of two pairs in two different linkage groups.

Similarly, the pseudo-testcross method was repeated with lod thresholds every 0.5 from 5.0 to 7.0. The number of incorrect markers at each lod thresholds is shown in table 1. A threshold of 7.0 resulted in 33 additional markers having incorrect phases compared to the original threshold of 4.7. Furthermore, every increase in the lod threshold resulted in at least one additional erroneous marker.

3.3 Pedigree size

To test the sensitivity of both methods in the presence of less data, subsets of the original grapevine data set were created with only a quarter, half, and three quarters of the original progeny. The results from both methods are shown in table 2. In the case of the pseudo-testcross method, the number of incorrect markers includes the discarded marker which was

Number of progeny	Incorrect markers with Phasing	Incorrect markers with pseudo-testcross
94	1	9
70	3	16
47	14	49
24	36	164

Table 2: The accuracy of Phasing and pseudo-testcross using different sized subsets of the individuals in the grapevine data set.

linked in the original output from Phasing. As the number of progeny decreases, both methods become less accurate, but the gap between the two increases.

Of course, pedigrees with fewer than 60 individuals are not very common in genetic mapping, because with so little data it would be difficult to perform any kind of accurate analysis, not just phase determination. Nonetheless, the results with small pedigrees serve to demonstrate that with less informative data sets, the accuracy of the two methods diverges.

3.4 Simulations

The simulated pedigrees consisted of 94 progeny and 20 linkage groups of equal sizes, with 2% missing data and a 1% error rate. The distribution of marker types is given in table 3. The proportion of $A0\times A0$ markers was made unrealistically high because it is the least informative segregation type, and thus, it provides the best test of the sensitivity of the algorithms. Three pedigrees were simulated with varying marker density and the accuracy of each of the two methods is shown in table 4. Again, Phasing was universally more accurate than the pseudo-testcross method. With the exception of three errors in the pseudo-testcross results on the 5 cM pedigree, all the errors were in the $A0\times A0$ markers. Thus, for both methods, the most difficult segregation type was $A0\times A0$.

Segregation type	Proportion
$AB\times CD$	10%
$AB\times AA$	25%
$AA\times AB$	25%
$AB\times AB$	25%
$A0\times A0$	15%

Table 3: Relative proportions of different marker types in the simulated data sets.

4 Discussion

In every test, Phasing was more accurate than the pseudo-testcross method. In pedigrees with many progeny, a high density of markers, and few $A0\times A0$ markers, the difference in accuracy was relatively small, but whenever any of these parameters was less than ideal, Phasing's greater sensitivity was more apparent.

The parameter T in equation 2 was introduced in order to make the maximization problem tractable, but it also makes the solution potentially less accurate. An alternative would be to find an approximate maximal solution with $T = 0$, for example, by using a stochastic search, rather than an exact maximal solution with $T > 0$. However, the degree of approximation in equation 2 is explicitly parameterized by the value of T, meaning that a more difficult data set would require a higher value of T. In contrast, the degree of sub-optimality of an approximate solution on a given data set is unknown. Even though experiments can validate the accuracy in general, a particularly difficult data set might violate the assumptions and produce a highly sub-optimal solution without any warning.

Furthermore, the results suggest that the accuracy does not depend greatly on the value of T. On the grapevine pedigree, the solution did not vary greatly for values of T between 3.4 and 7.0, so it seems that the theoretical advantages of lower values of T are

Distance between Markers	Number of markers	Pseudo-testcross error rate	Phasing error rate
5 cM	400	3.25%	0.75%
2 cM	1000	1.2%	0.4%
1 cM	2000	0.85%	0.0%

Table 4: Results of pseudo-testcross and Phasing on simulated data sets.

not so large in practice.

On the other hand, there is no advantage to using a higher value of T. The false apparent linkage between markers in different linkage groups which occurs at lower values of T does not make the resulting phases less accurate because the relative phases between different linkage groups do not matter.

In fact, the algorithm as described does not explicitly recognize that the markers belong to distinct linkage groups. In contrast, most genetic analyses begin by dividing the markers into linkage groups and then analyze each linkage group individually. While the phases of individual linkage groups would have a smaller search space and so equation 3 would have fewer terms, it would add the additional step of finding the linkage groups in phase-unknown data. Nevertheless, for larger data sets than the one used in this paper, the linkage groups may need to be analyzed individually in order to be able to find a solution with a sufficiently low threshold T.

The current implementation only deals with pedigrees consisting of the progeny of a single cross, but the same algorithm can be extended to some other, more complex pedigree types. Accurate phase determination inherently requires many offspring, which rules out many potential applications. With enough offspring the analysis can always be carried out considering each cross and its progeny in isolation, but this discards information. For example, CEPH-like pedigrees can be analyzed by ignoring the grandparents, which is obviously less efficient because the grandparents' genotypes can sometimes fix the phases. On the other hand, the algorithm could be extended to assume from the beginning those phases which are unambiguous from the pedigree structure and only solve for the remaining unambiguous phases.

In conclusion, while not perfectly accurate, Phasing is quite accurate for the most common pedigrees and represents an improvement over existing methods in both accuracy and convenience.

5 Acknowledgments

The author would like to thank Michela Troggio for introducing me to this problem and providing data for testing and Alexander Gutin for many suggestions with the manuscript.

References

[1] A. F. Adam-Blondon, C. Roux, D. Claux, G. Butterlin, D. Merdinglu, and P. This. Mapping 245 SSR markers on the *Vitis vinifera* genome: A tool for grape genetics. *Theor. Appl. Genet.*, 109(5):1017–1027, September 2004.

[2] J. Dausset, H. Cann, D. Cohen, M. Lathrop, J. M. Lalouel, and R. White. Centre d'etude du polymorphisme humaine (CEPH): Collaborative genetic mapping of the human genome. *Genomics*, 6(3):575–577, March 1990.

[3] Simon de Givry, Martin Bouchez, Patrick Chabrier, Denis Milan, and Thomas Schiex. CarTaGène: Multipopulation integrated genetic and radiated hybrid mapping. *Bioinformatics*, 21:1703–1704, 2005.

[4] Robert C. Elston and John Stewart. A general model for the genetic analysis of pedigree data. *Hum. Hered.*, 21(6):523–524, 1971.

[5] M. Stella Grando, D. Bellin, K. J. Edwards, Carlo Pozzi, M. Stefanini, and Riccardo Velasco. Molecular linkage maps of Vitis vinifera L. and Vitis riparia Mchx. *Theor. Appl. Genet.*, 106(7):1213–1224, May 2003.

[6] Dario Grattapaglia and Ronald Sederoff. Genetic linkage maps of *Eucalyptus grandis* and *Eucalyptus urophylla* using a pseudo-testcross: Mapping strategy and RAPD markers. *Genetics*, 137:1121–1137, August 1994.

[7] Phil Green, Kathy Falls, and Steve Crooks. *CRI-MAP Documentation. version 2.4*, 1990.

[8] Eric S. Lander and Philip Green. Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA*, 84(8):2363–2367, April 1987.

[9] M. A. Lodhi, M. J. Daly, G.-N. Ye, N. F. Weeden, and B. I. Reisch. A molecular marker based linkage map of Vitis. *Genome*, 38:786–794, August 1995.

[10] S. Riaz, G. S. Dangl, K. J. Edwards, and C. P. Meredith. A microsatellite marker based framework linkage map of *Vitis vinifera* L. *Theor. Appl. Genet.*, 108:864–872, March 2004.

[11] Warren M. Snelling, Mathieu Gautier, John W. Keele, Timothy P. L. Smith, Roger T. Stone, Gregory L. Bennett, Naoya Ihara, Akiko Ataksuga, Haruko Takeda, Yoshikazu Sugimoto, and André Eggen. Integrating linkage and radiation hybrid mapping data for bovine chromosome 15. *BMC Genomics*, 5(1):77, 2004.

[12] Piet Stam. Construction of integrated genetic linkage maps by means of a new computer package: JoinMap. *Plant J.*, 3:739–744, 1993.