

Association Mapping of Complex Diseases with Ancestral Recombination Graphs: Models and Efficient Algorithms

Yufeng Wu

Department of Computer Science
University of California, Davis
Davis, CA 95616, U.S.A.
wuyu@cs.ucdavis.edu

Abstract. Association, or LD (linkage disequilibrium), mapping is an intensely-studied approach to gene mapping (genome-wide or in candidate regions) that is widely hoped to be able to efficiently locate genes influencing both complex and Mendelian traits. The logic underlying association mapping implies that the best possible mapping results would be obtained if the genealogical history of the sampled individuals were explicitly known. Such a history would be in the form of an “ancestral recombination graph (ARG)”. But despite the conceptual importance of genealogical histories to association mapping, few practical association mapping methods have explicitly used derived genealogical aspects of ARGs. Two notable exceptions are [35] and [23].

In this paper we develop an association mapping method that explicitly constructs and samples minARGs (ARGs that minimize the number of recombinations). We develop an ARG sampling method that provably samples minARGs *uniformly* at random, and that is practical for moderate sized datasets. We also develop a different, faster, ARG sampling method that still samples from a well-defined subspace of ARGs, and that is practical for larger sized datasets. We present novel efficient algorithms on extensions of the “phenotype likelihood” problem, a key step in the method in [35]. We also prove that computing the phenotype likelihood for a different natural extension of the penetrance model in [35] is NP-hard, answering a question unresolved in that paper. Finally, we put all of these results into practice, and examine how well the implemented methods perform, compared to the results in [35]. The empirical results show great speed ups, and definite but sometimes small, improvements in mapping accuracy. Speed is particularly important in doing genome-wide scans for causative mutations.

1 Introduction

One type of genetic disease, or more generally any phenotype (observable trait), is caused by a single mutation at a single locus, and that mutation has “high penetrance”, meaning that the probability of the trait given the mutation is very

large. Sometimes this type of disease is called “Mendelian”. In contrast, “complex traits” can originate independently at many different loci; or a combination of mutations is required to create a phenotype; or different combinations of mutations can create the trait; or the penetrance of the mutations may be low. Understandably, although many Mendelian traits have been mapped quite successfully, mapping the genetic origin of complex traits remains a very challenging problem.

Association, or LD (linkage disequilibrium), mapping is a current, intensely-studied approach to gene mapping (genome-wide or in candidate regions) that is widely hoped to be able to efficiently locate genes influencing both complex and Mendelian traits. Indeed, one of the major motivations behind the international HapMap project [16, 17] is to provide SNP (single nucleotide polymorphism) data from several populations, at a density of about one SNP per one to five Kb, to facilitate association mapping (in humans). The association mapping approach uses (sparse) data obtained from a number of *unrelated* individuals in a population, looking for sites, or small regions, whose states strongly discriminate between those individuals (called “cases”) with the trait of interest, and those without it (called “controls”). Association mapping relies on the assumption that the cases (or a significant fraction of them) share a genealogical history that is distinct from the history of the controls, and that over time, meiotic recombination has shortened the shared region(s) containing the causative mutation(s). It follows from these assumptions that SNP sites near a causative mutation will have states (alleles) that more highly correlate with the trait of interest than do sites that are far from a causative mutation, and this is the general basis for association mapping. The following papers provide good overviews and discussions of association mapping [28, 5].

The logic behind association mapping implies that the best possible mapping results would be obtained if the true genealogical history was explicitly known for the cases and controls. Such a history would be in the form of an “ancestral recombination graph (ARG)” [7, 26, 13], also called a “phylogenetic network” in [9, 8]. The true ARG would explicitly show all the ancestral relations, the mutations and the recombinations that lead to the extant SNP sequences of the sampled individuals, starting from some ancestral SNP sequence. Quoting from a recent paper (also see [23]):

Unless we have the actual disease variants in our marker set, the best information that we could possibly get about association is to know the full coalescent genealogy of our sample at that position. If we knew this, the marker genotypes would provide no extra information; ... [35]

It is important to note that the concept of ARG in this paper is synonymous with phylogenetic network, which is about a network showing ancestral relations among samples. This use of the term “ARG” is similar to the usage of this term in [23], and is *different* from a stochastic process called coalescent-with-recombination. Even with this restriction, an ARG is still very informative about the genealogical history of the samples.

Despite the conceptual centrality of the genealogical history, few association mapping methods have tried to explicitly deduce or exploit the underlying genealogical histories to map complex traits, particularly with recombination. Initial work on association mapping with ARGs by a full coalescent likelihood approach suggests that this is indeed a very challenging problem [19], and most existing genealogy-based mapping approaches [33, 22, 24, 27] make some approximations (i.e. not using a full genealogical network) in modeling recombination.

Recently, however, a few papers have developed association mapping methods that explicitly try to exploit recombination or some aspects of the “underlying ARG space”, that is, the set of ARGs that can generate a given input set of SNP sequences. Two papers, the first published in 2005 by Zollner and Pritchard [35], and the second by Minichiello and Durbin [23], are the most highly developed examples of these efforts.

The method of Zollner and Pritchard [35] explicitly uses some inferred information on recombination for genealogy inference, although it does not generate full ARGs. The method uses a rigorous stochastic framework and disease model to map certain kinds of complex traits. The basic strategy in [35] is to generate, and average over, some information from many samples of the ARG space for the given data. In particular it generates (independent) subtrees embedded in the ARGs, at different loci. Each such tree describes how the SNP sequences, restricted to an interval of SNP sites, could have evolved. The quality of a subtree is assessed using a rigorous statistical model (detailed later). A locus where many generated subtrees have significant scores is then deduced as being near a site that is causative for the trait.

The Zollner and Pritchard paper is an important advance because it defines a formal disease model, and it uses a rigorous likelihood approach to evaluate the significance of the mapping results. Moreover, it considers complex phenotypes showing “allelic heterogeneity”, where the genetic basis for the trait can be a mutation at one of several different sites, but the sites are located close to each other. This is the case for example for BRCA1, or for mutations causative for the ability to metabolize lactose in adults. However, the implemented method (based on Markov Chain Monte Carlo) is very slow in practice, does not guarantee proper mixing, and does not use the full ARG model. Moreover, the disease model is somewhat limited and it would be desirable to extend it in several natural directions.

More recently, Minichiello and Durbin [23] developed an association mapping method that is similar to that in [35] at a high-level, but quite different in detail. In [23], full “plausible” ARGs are explicitly generated by using heuristics that allow rapid computation. There is no precise definition of what a “plausible” ARG is, although the algorithm tries to locally reduce the number of recombinations used, and can be viewed as producing an approximation to a “minARG” [29], i.e., an ARG that globally minimizes the number of recombinations used to generate the SNP sequences (in a model detailed below). There is no characterization of the sampling bias that is caused when ARGs are created in this way.

Our paper is centrally motivated by the paper of Zollner and Pritchard [35]. Our paper addresses computational, and some statistical, challenges from [35], with results concerning all the essential steps of the method in [35]. Essentially, we show that the statistical approach in [35] can be sped up and made practical when full minARGs (or near-minimum ARGs) are sampled. We adopt the disease model introduced in [35] and present new results in evaluating the “phenotype likelihood”, used to assess the significance of a “marginal tree” (defined below). However, some parts of our method are more similar to the method in [23], and so some of our results also relate to that paper.

2 Definitions and Background

A *single nucleotide polymorphism (SNP)* is a single nucleotide site where exactly two (of four) different nucleotides occur in a large percentage of the population. A *genotype* comes from a pair of *haplotypes*. As in [35] we assume that haplotypes can be determined from sampled genotypes, and to simplify the exposition, we just say that haplotypes are sampled in the population. The set of haplotypes sampled from a population is denoted by M , where M has n haplotypes (rows) and m sites (columns). We assume at most one mutation in any sampled SNP site in the evolution of the haplotypes, which is supported by the standard “infinite sites model” in population genetics [13, 14]. This is particularly justified in the context of association mapping where the time-scale of interest is short enough that two mutations at any single site are unlikely. In addition to mutation, haplotypes evolve by (meiotic) recombination. Recombination takes two equal length sequences (haplotypes) and produces a third sequence of the same length consisting of some *prefix* of one sequence, denoted P , followed by a *suffix*, of the other sequence, denoted S . The changeover point is called the “crossover point” or “breakpoint”.

The evolutionary history of a set of haplotypes M , that evolve by mutations and recombinations is displayed on a rooted, directed acyclic graph called an “Ancestral Recombination Graph (ARG)” [7] (also in [26, 13]), or a “Phylogenetic Network” in [9, 8]. An example of an ARG is shown in Figure 1(b); A formal definition of an ARG is given in [9, 8]. An ARG that derives a set of sequences M and *minimizes* the number of recombinations assuming at most one mutation per site is called a “minARG” [29]. The problem of finding a minARG for M , or even determining the number of recombinations in it, is NP-hard [34, 3], but there are methods constructing minARGs for moderate-size haplotype data [29, 21]. Other methods construct minARGs with some structural constraints [8–10]. We can also efficiently compute close upper bounds [11, 12, 31] and lower bounds [15, 25, 30, 1, 2, 31] on the number of recombinations in a minARG.

The marginal trees of an ARG

Let N denote an ARG for M . The following crucial observation is central in the methods in [35] and [23] and in our method. For any site x , the full evolutionary

history of the states of site x in the sequences M , is completely represented by a *subtree* T_x of N , which can be extracted from N by removing, at each recombination node v in N , one of the two directed edges entering v . In particular, suppose b is the breakpoint for the recombination at v ; then remove the edge into v from the node labeled with the sequence P (providing the prefix for the recombination), if site x is to the right of b ; otherwise remove the edge into v from the node labeled S . The resulting subtree, T_x , is a rooted directed tree that details how each of the sequences in M obtained their polymorphism value at x . An example is given in Figure 1. Tree T_x is called a *marginal tree*. The next critical point is that if no recombination in N has occurred at a breakpoint between sites x and y , then the marginal trees T_x and T_y are identical. Hence, there is a *single* well-defined marginal tree for each interval between successive breakpoints in N (along the linear ordering of the polymorphic sites), and also for the two intervals before the first, and after the last, breakpoints.

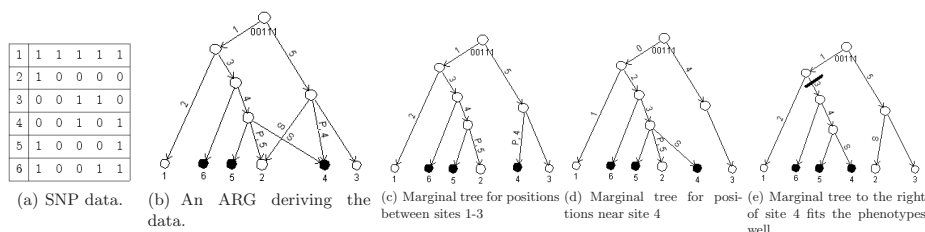


Fig. 1. An example of the general association mapping method using ARGs. Figure 1(a) shows the input haplotypes M , where rows 1-3 are controls and rows 4-6 are cases. Figure 1(b) shows an ARG for the haplotypes in Figure 1(a), with 00111 as root sequence. A site mutation changes the state from that at the root to the opposite state. Leaves are labeled with row numbers in the input matrix. Figures 1(c) is the marginal tree embedded in the ARG at positions between site 1 and 3. Figure 1(d) is the marginal tree near site 4. Figure 1(e) is the marginal tree to the right of site 4. Note this tree shows a cut of an edge that clearly separate cases and controls.

3 The High-level Strategy

Our high-level strategy involves sampling ARGs from the ARG space; then constructing the marginal trees for each sampled ARG; and then assessing the statistical significance of the marginal trees given the observed cases and controls, and the disease model. To get the intuition behind this assessment, suppose the disease is Mendelian. In that case a significant marginal tree is expected to have an edge, such that there is a “larger than random” number of cases in the leaves of the subtree below the edge, and few controls. A locus contained in marginal trees that have high significance is then deduced as being near a site that is causative for the trait. This basic idea is adopted in [23]. The general method,

in the Mendelian case, is illustrated in Figure 1, where the third marginal tree corresponding to sites to the right of site 4, contains an edge that perfectly separates the cases from the controls. Therefore, the tree to the right of site 4 *fits* the observed phenotypes well. If no further sampling were done, we might then conclude that the genomic region near site 5 is the most likely to contain the causative mutation for the trait. Now, for complex diseases, the assessment of the significance of a marginal tree is more involved. Then we need to compute the probability, given the disease model, that the observed cases and controls would have been derived on that marginal tree. If the disease model specifies low penetrance or multiple causative mutations, then we would not expect a perfect separation of the cases and controls as in Figure 1.

The above high-level strategy is in the same spirit as some parts of the methods in [35] and [23], but our method differs from, extends and sometimes outperforms those methods in several important ways. Most importantly, in contrast to both earlier methods, our methods explicitly compute minARGs, or ARGs that (empirically) use a number of recombinations close to the global minimum, rather than other ARGs from the ARG space. There are several reasons we want to generate minARGs (or near minimum ARGs) rather than other ARGs from the ARG space. First, it is currently believed that in the human genome (and perhaps others) there are many regions (haplotype blocks) where the recombination rate is low, but not zero. In those regions, we expect that minARGs reflect the true genealogical history better than an ARG with many extra recombinations. Also, the method in [23] has implicit (but not rigorous) rules for reducing the number of recombinations used, and the use of minARGs better formalizes that implicit effect.

Phenotype likelihood is defined as, given a marginal tree T_x at position x with leaf labels, the probability $Pr(\Phi|X = x, T_x)$ of the observed phenotypes Φ of the leaves being generated on T_x according to the following disease model assuming disease mutations occur near x [35]. Here Φ is the collection of case and control status for each sample sequence. The basic *disease model* that we adopt in this paper is the one introduced by Zollner and Pritchard [35].

- The disease loci are not sampled, i.e. are not in the sampled SNP sites.
- Phenotypes are determined by mutations at disease loci and the disease penetrance. There may be multiple independent disease mutations, but these mutations occur relatively close together, so that they may all occur on a single marginal tree.
- There are two alleles M_0 (wild-type) and M_1 (mutant) at a disease locus. Mutations M_0 to M_1 occur at edges of the marginal tree according to Poisson process with a rate of $\nu/2$. There is *no* mutation from M_1 to M_0 . Furthermore, mutations on different edges of a marginal tree occur independently.
- Multiple mutations on the same haplotype have the same effect on phenotypes as a single mutation.

The concept of *penetrance* is important to this paper. Zollner and Pritchard uses the *haploid penetrance* model to specify the effect of alleles M_0, M_1 on

phenotypes: $P_{\phi,m}$ is the probability of a haplotype exhibiting phenotype ϕ for wild-type haplotype ($m = 0$) or mutant haplotype ($m = 1$). Here, $\phi \in \{A, C\}$ (i.e. case or Control). Since $P_{\phi,m}$ is for a single sequence (haplotype), we call it haploid penetrance. In practice, penetrances are often not known. Zollner and Pritchard used a numerical integration approach, averaging over a grid points of penetrance (e.g. an evenly spaced 20 by 20 grid with range $[0.0, 1.0]$, where each point represents a possible penetrance $P_{A,0}$ and $P_{A,1}$). Thus in the following, we assume penetrance is known when we compute phenotype likelihood.

Suppose we are given a (rooted) tree T_x . Let M be a binary vector with one bit for each edge, indicating whether the corresponding edge has at least one disease mutation or not. Then,

$$Pr(\Phi|X = x, T_x) = \sum_M P_{A,0}^{n_{A,0}} P_{A,1}^{n_{A,1}} P_{C,0}^{n_{C,0}} P_{C,1}^{n_{C,1}} Pr(M|x, T_x)$$

where $n_{\phi,m}$ = number of sequences in the sample showing phenotype ϕ who have mutation state m , and $Pr(M|x, T_x)$ is the probability of the edge mutations specified by M in T_x , which can be easily computed. Zollner and Pritchard use the *Peeling algorithm* [6] to efficiently compute phenotype likelihood with haploid penetrance. Understanding this is important for our new results on phenotype likelihood in Section 5. Refer to [35] for more details.

4 Sampling Ancestral Recombination Graphs

4.1 Uniform sampling of minARGs

Now we present two methods for sampling ARGs given a set of sequences M . We first present a method to count the number of minARGs. With a simple modification, the method can be turned into a uniform sampler of minARGs.

We begin by reviewing the *self-derivability* problem originally studied in [31]. In the following, we assume that M does not contain two identical sequences. Often an ARG may derive sequences that are not present in input data M . We call these sequences *Steiner* sequences. The self-derivability problem is to decide whether there is an ARG N deriving M (assuming at most one mutation per site) which only contains the input sequences in M . That is, N contains no Steiner sequences. We call such an ARG, if it exists, a *self-derived* ARG. We first describe algorithms for data with self-derived ARGs, and then extend to more general case. Lemma 1 is a simple extension of a result in [31] and we omit the proof here.

Lemma 1. *A self-derived ARG is also a minARG.*

Here, we give an algorithm (Algorithm 1) for counting the number of self-derived ARGs for M . The algorithm runs in $O(2^n + n^3m)$ time. Two ARGs are different if they derive a different set of sequences. Moreover, the nodes (i.e. sequences) in an ARG are derived in a particular linear time order. That is, for

any two nodes, we know which corresponding sequence is derived earlier. We consider two ARGs to be different if the derivation order of the sequences in them are different, even if they are topologically identical and derive the same set of sequences. This total time-order property is quite convenient to avoid over-counting (as explained later). Also, the time-ordering suggests ways to determine edge lengths for the edges in an ARG, and this will be useful for the phenotype likelihood computation, discussed later. Moreover, true ARGs are total-ordered since genealogical events are time-ordered.

For each subset $S \subseteq \text{Rows}(M)$, we define $N[S]$ as the *number* of the minARGs deriving sequences in S (and deriving no other sequences). Therefore, $N[\text{Rows}(M)]$ is equal to the total number of self-derived ARGs that derive haplotype matrix M . For a subset of rows S and a single row $r \notin S$, we denote $D(S, r)$ as the total number of ways of deriving r by sequences in S through an unused mutation or a recombination. By "unused mutation" we mean a mutation at a site where all sequences in S have the same states (i.e. either all 0 or all 1) and different from that of r . Note that if r is derived by an unused mutation, this is only one way of deriving r and no two sequences in S can recombine to derive r . Similarly, when r can be derived through recombination of two sequences in S , r can not be derived through an unused mutation from a sequence in S . Thus, there are the following mutually exclusive situations:

1. If a sequence in S can derive r by an unused mutation, then $D(S, r) = 1$.
2. If two sequences in S can derive r by a recombination, then $D(S, r) \geq 1$.
3. Otherwise, $D(S, r) = 0$.

Algorithm 1

1. For each row $r \in M$, set $N[\{r\}] \leftarrow 1$.
2. Set $sz \leftarrow 1$.
3. while $sz < n$
 - 3.1 $sz \leftarrow sz + 1$
 - 3.2 For each subset of rows $S \subseteq \text{Rows}(M)$ and $|S| = sz$, initialize $N[S] \leftarrow 0$.
 - 3.2.1 For all $r \in S$, such that (a) $N[S - \{r\}] \geq 1$, and (b) $D(S - \{r\}, r) \geq 1$, then $N[S] \leftarrow N[S] + N[S - \{r\}] \times D(S - \{r\}, r)$.

The correctness of Algorithm 1 is easy to establish from the total time-ordered property. The key observation is that since the ARG is fully time ordered, each way of choosing r generates a different ARG. Intuitively, picking r means we choose to derive r *immediately* after the sequences in $S - \{r\}$ in the ARG. Two ARGs generated by picking r_1 and r_2 respectively at the same stage are different because the time order of r_1, r_2 is different.

Now we show that Algorithm 1 can be converted to a uniform sampler of minARGs for data M when M is self-derivable. This is presented in the following.

Algorithm 2

1. First count the total time-ordered minARGs using Algorithm 1.
2. Initialize S , the set of current underderived rows, to $\text{Rows}(M)$. Do:

- 2.1 Choose a sequence r from S as the last sequence in S to derive, with probability $\frac{N(S-\{r\}) \times D(S-\{r\}, r)}{N(S)}$.
 - 2.2 Choose uniformly at random (and remember) a derivation way (i.e. either through a mutation or a recombination) for r from total $D(S-\{r\}, r)$ possible ways of derivation.
 - 2.3 Let $S \leftarrow S-\{r\}$. Go to Step 3 if $|S| = 1$ (i.e. the only remaining sequence is the root sequence). Otherwise, continue on step 2.1.
3. Construct an ARG (starting from the root) according to the (reverse) order for each sequence r and the chosen way of deriving r .

It is easy to show that the above algorithm is indeed a uniform sampler of self-derived minARGs. Here is the intuitive idea behind this method. We construct a uniformly sampled time-ordered minARG backwards in time. That is, we decide how to derive the last sequence first. For a given data, we choose a sequence r as the last sequence to derive in the minARG with probability equal to the ratio of the number of minARGs with r as the last sequence to the total number of minARGs. This ensures that the last sequence is picked uniformly. We pick the rest of sequences uniformly backwards in time. Thus, the generated ARG is sampled uniformly. We remark that the (exponential) set-up time for the uniform sampling is the dominant portion of the running time. Note however that once the counting is performed, sampling of an ARG takes $O(n^2)$ time.

A remaining issue is that the above uniform sampling method only works for a special type of data M , namely the data where M is self-derivable. The method can be extended to handle general data for moderate-sized datasets as follows. When the number of needed Steiner sequences and the number of candidate Steiner sequences are small, we can simply add Steiner sequences to M in order to make the expanded dataset self-derivable. This and several other ideas that we have implemented make uniform sampling of minARGs practical in a range of data we report in Section 6. Recall also that association mapping is often done on candidate regions or on windows in a genome, and these also fall in the range of practicality for minARG generation. The efficiency of our minARG sampling method depends largely on the number of haplotypes (and number of Steiner sequences needed) of M . Our experience indicates that minARGs can often be found (within practical amount of time) for data with up to 30 haplotypes when the data is self-derivable, and up to 20 haplotypes when the number of sites is small and one or two Steiner sequences are needed.

Remarks. The above uniform sampling can be extended to weighted sampling, where larger weights are assigned to more likely operations. Weighted sampling may improve the mapping accuracy. We omit the details here.

4.2 ARG sampling for larger data

The performance of the uniform sampling method degrades with the increase of the number of haplotypes in M or the number of needed Steiner sequences. To sample ARGs for larger data, we need heuristic sampling methods.

The efficient ARG sampling method presented here samples a special type of ARGs, where sequences are derived by a derivation pathway. Constructing a single ARG by a derivation pathway has been previously used in [31], and implicit application of the pathway is also used in [20] for estimating recombination rate. In this paper, we develop an ARG sampling method based on minimum pathway. Minimum pathway is a way of deriving a new sequence from a set of derived sequences using the fewest recombinations. Thus, we also explicitly reduce the number of recombinations here. We also sample uniformly derivation paths from minimum pathways. We call the ARGs constructed from pathways as pathway ARGs, and this sampling method *pathway sampling*.

The detailed description of the pathway ARG sampling method together with a uniform sampling method of a derivation path from the minimum pathway for a fixed derivation order is deferred to the full version of this paper.

5 Phenotype Likelihood

5.1 More on phenotype likelihood

An alternative to the the phenotype likelihood (denoted as PL and described in Section 3) in [35] is that, instead of summing over all possible subsets of mutated edges, we seek *one* subset of edges in the marginal tree where disease mutations occur, which *maximizes* the probability of the observed phenotypes caused by the chosen mutations [4]. Due to the lack of a better name, we name the maximized probability as maximum phenotype likelihood (MPL). More precisely, MPL is equal to $MAX_M(Pr(\Phi|M, X = x, T_x) * Pr(M|X = x, T_x))$, where M is a vector indicating on which tree edges mutations occur. It is easy to demonstrate an efficient procedure (details omitted) for computing maximum likelihood with haploid penetrance by dynamic programming (a variant of the peeling algorithm used in [35]). Initial experiments show, however, mixed results of using MPL as the scoring scheme. Thus, in this paper, we adopt the original PL scheme as the scoring scheme.

5.2 Expected phenotype likelihood

An important computational problem for a statistical method is assessing statistical significance of the results. For the phenotype likelihood problem, a natural question to ask is whether the given phenotypes are indeed caused by disease mutations (i.e. the alternative model) or just some random noise (i.e. the null model). Imagine that we randomly permute phenotypes of leaves in a given tree (i.e. without changing the number of cases). A commonly used scheme to assess statistical significance is to compute a P-value, which is the adopted method in [35]. In practice, computing the P-value is often through permutation tests. Permutation tests may not give accurate result and are time-consuming, and may be the bottleneck of whole genome scan for trait mapping [35].

Besides the P-value, other statistics may provide hints on statistical significance, including, for example, the expected phenotype likelihood. It might

be possible to develop another scheme for assessing significance involving the expected likelihood. Unlike the permutation tests used in [35], our method computing the expected phenotype likelihood is an exact method, running in polynomial-time and fully deterministic. In the following, we show that the expected value of phenotype likelihood with haploid penetrance can be efficiently computed. With some small modifications, we can also compute variance of phenotype likelihood with a polynomial-time algorithm (details omitted). For simplicity, we assume the marginal tree T is binary. Note that if the given T is not binary, we can easily transform T to a binary tree T' without changing the phenotype likelihood [4].

We define $L_r(s, m, k)$ to be expected (randomized) phenotype likelihood for the subtree under node (i.e. sequence) K_s where node K_s has disease mutation state m and the subtree contains exactly k case haplotypes. Recall that m is either 0 (K_s is a wild-type) or 1 (K_s is a mutant). The base case when K_s is a leaf is easy. We have, $L_r(s, 0, 0) = 1.0 - P_{A,0}$, $L_r(s, 1, 0) = 1.0 - P_{A,1}$, $L_r(s, 0, 1) = P_{A,0}$, and $L_r(s, 1, 1) = P_{A,1}$.

Now consider an internal node K_s with k case haplotypes under K_s . Denote the number of leaves (both cases and controls) under K_s as $k_{t,s}$. Denote the two children of K_s as K_{s_l} and K_{s_r} . Denote μ_l is the probability of at least one mutation occurs at edge from K_s to K_{s_l} . Similarly, μ_r is defined for K_{s_r} . There are up to $O(k)$ different way of splitting the k cases into two subtrees under K_{s_l} and K_{s_r} . Suppose in one way of splitting, we have k_1 cases in subtree under K_{s_l} and $k - k_1$ cases in subtree under K_{s_r} . The probability of such split is equal to the probability of a randomly chosen k_{t,s_l} balls from total $k_{t,s}$ balls (with k black balls) such that k_1 black balls are chosen. This is equal to:

$$P_s(s, k, k_1) = \frac{\binom{k}{k_1} \binom{k_{t,s}-k}{k_{t,s_l}-k_1}}{\binom{k_{t,s}}{k_{t,s_l}}}$$

Therefore, we have the following recursions (whose proof is omitted) :

$$L_r(s, 1, k) = \sum_{k_1} P_s(s, k, k_1) L_r(s_l, 1, k_1) L_r(s_r, 1, k - k_1)$$

When $d = 0$, we need to consider the probability of edge mutation as well.

$$L_r(s, 0, k) = \sum_{k_1} P_s(s, k, k_1) * ((1 - \mu_l) L_r(s_l, 0, k_1) + \mu_l L_r(s_l, 1, k_1)) * ((1 - \mu_r) L_r(s_r, 0, k - k_1) + \mu_r L_r(s_r, 1, k - k_1))$$

Note that the expected phenotype likelihood is precisely $L_r(r, 0, n_c)$, where K_r is the root of the tree and n_c is the number of cases. The above recursion can be easily implemented in a dynamic programming algorithm with $O(n^3)$ running time.

5.3 Diploid penetrance

Zollner and Pritchard used haploid penetrance in phenotype likelihood computation. Since we are mostly interested in diploid samples, diploid penetrance, rather than haploid penetrance, seems more natural. A diploid sample contains *two* haplotypes, and its phenotype is decided by the *joint* mutation status of the two haplotypes. In diploid penetrance, we have $P_{\phi,00}$, which is the probability of a diploid sample exhibiting phenotype ϕ if *both* of its haplotypes are wild-type haplotypes. Similarly, $P_{\phi,01}$ (resp. $P_{\phi,11}$) is the probability of a diploid sample exhibiting phenotype ϕ if exactly one (resp. none) of its haplotypes is wild-type haplotype. An important question stated but unanswered in Zollner and Pritchard [35] is how to efficiently compute phenotype (denoted $Pr_d(\Phi|X = x, T_x)$) likelihood using diploid penetrance model.

The main difference between $Pr_d(\Phi|X = x, T_x)$ and $Pr(\Phi|X = x, T_x)$ is that the diploid likelihood considers a diploid individual as a single entity rather than two haplotypes as in haploid likelihood. Similar to the haploid penetrance case, we can define the maximum likelihood problem with diploid penetrance. For easy reference, we name the problem of computing $Pr_d(\Phi|X = x, T_x)$ Diploid-Phenotype-Likelihood or DPL. We name the problem regarding maximum likelihood as Max-Diploid-Phenotype-Likelihood or MDPL. Theorem 1 says that diploid likelihood problems are NP-hard, whose proof is deferred to the full version of this paper.

Theorem 1. *The MDPL and DPL problems are both NP-hard.*

6 Experimental Results

The described methods are implemented using C++ in an association mapping program, which we name as Trait Mapping tool with ARG (TMARG). For testing statistical significance of phenotypes, TMARG uses the PL scheme with haploid penetrance as the scoring scheme. Different from LATAG (program developed in [35]), we take maximum of PL values over penetrance grid points, rather than taking average. Initial experiences show that taking maximum slightly improves mapping accuracy. However, taking maximum also appears to give less repeatable mapping results.

TMARG takes a matrix of haplotypes or phase-known genotypes and their phenotypes (i.e. case/control status) as input, and provides point estimate for the complex trait loci. TMARG supports both uniform sampling of minARGs in a sliding window, and pathway ARG sampling for the *entire* data. TMARG currently only allows data consisting of binary SNPs. For unphased or noisy data, we suggest to first preprocess the data using haplotype inference programs, such as PHASE [32]. To evaluate the effectiveness of our program, we test with both real biological data and simulated data. We compare TMARG with both LATAG and MARGARITA (the program developed in [23]). When running MARGARITA, we sample 50 ARGs and perform 10000 permutations for each data.

The first data is the Cystic Fibrosis (CF) data [18], which has been analyzed by many association mapping methods. It contains 23 binary markers over 1.8 Mb. There are 94 disease haplotypes and 92 control haplotypes. The most common mutation is located 885 Kb from the left end of the region. We use program PHASE 2.1 [32] to impute missing data. The uniform sampling scheme gives the point estimate at 1096 Kb, while the pathway sampling scheme gives 915 Kb. For each method, we perform 50 independent runs, while in each run we sample 50 genealogies. The reported results are the consensus point estimates over the 50 runs. The LATAG’s point estimate is at 867 Kb, while MARGARITA’s point estimate is at 870 Kb.

The second data contains 50 simulated datasets used in Zollner and Pritchard [35], which we call ZPS data. These data were intended for testing effectiveness of gene mapping methods regarding complex traits. Each ZPS data contains 30 diploid cases and 30 diploid controls (with known phases). Each data typically contains between 45 to 65 binary markers. The generation of these data essentially follows from the disease model in Section 5. Typically 10-25 disease mutations (including many redundant) are generated for each data. One reason that these data may not be easy for mapping is that only part (10 to 33 among 60) of case haplotypes do actually carry disease mutations while some (0 to 9 among 60) control haplotypes also carry disease mutations. Table 1 lists our mapping result using TMARG, comparing to LATAG/MARGARITA.

Table 1. Mapping for simulated data in Zollner and Pritchard [35]. We test both Uniform and Pathway sampling schemes. We measure the accuracy by the average point estimate error, standard error and percentage of data with point estimate within 0.1 cM distance from the true trait loci for the 50 datasets. The units of all the point estimates are cM. The results of TMARG are consensus from 10 independent runs, where each run sample 50 or 5000 genealogies.

	U	P	P	LATAG	MARGARITA
Sample Num	50	50	5000	50	50
Ave. Err.	0.184	0.180	0.166	0.19	0.229
Std. Err.	0.215	0.210	0.197	0.23	0.255
% < 0.1 cM	50%	50%	56%	54%	44%

Our simulation results show that TMARG is comparable with LATAG and MARGARITA for CF data and slightly outperforms the other two programs in accuracy for the ZPS data. Note that we only tested MARGARITA with one settings and its mapping result may change when using different settings (e.g. more permutations per data). We note that both uniform sampling and pathway sampling methods are comparable to MARGARITA in speed and are much faster than LATAG for the data we tested (when same number of samples are generated). For example, for the CF data, LATAG takes 8 hours for each run, while our sampling methods take a few minutes. As seen in Table 1, pathway sampling method appears to be slightly more accurate than the uniform sam-

pling method and sampling more ARGs per data may slightly improve mapping accuracy. On the other hand, uniform sampling appears to produce more repeatable results than the pathway sampling. We remark that more simulation tests on both real biological and simulated data are needed to further validate our proposed methods and compare our methods with LATAG/MARGARITA.

Acknowledgments. I am very grateful to my advisor, Dan Gusfield, for many invaluable discussions and for reading the manuscript carefully and making many good suggestions. I would like to thank Dan Brown for very fruitful and stimulating discussions on the phenotype likelihood problems. I thank Chuck Langley for making several important comments on the manuscript. I also thank Yun S. Song and Katie Pollard for helpful discussions. Work supported by grants CCF-0515278 and IIS-0513910 from National Science Foundation. Simulations are performed on a Linux cluster supported by NSF grant CNS-0224469.

References

1. V. Bafna and V. Bansal: The number of recombination events in a sample history: conflict graph and lower bounds. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, v.1, 78-90, 2004.
2. V. Bafna and V. Bansal, Inference about Recombination from Haplotype Data: Lower Bounds and Recombination Hotspots. *J. of Comp. Bio.*, v.13, p.501-521, 2006.
3. M. Bordewich and C. Semple: On the computational complexity of the rooted subtree prune and regraft distance. *Annals of Combinatorics*, v.8, p.409-423, 2004.
4. D. Brown: Private communications.
5. A. G. Clark: Finding genes underlying risk of complex disease by linkage disequilibrium mapping. *Current Opinion in Genetics and Development*, 13, p.296-302, 2003.
6. J. Felsenstein, Evolutionary trees from DNA sequences: a maximum likelihood approach, *J. Mol. Evol.*, v.17, p.368-376, 1981.
7. R. C. Griffiths and P. Marjoram: Ancestral inference from samples of DNA sequences with recombination. *J. of Comp. Bio.*, v.3, p.479-502, 1996.
8. D. Gusfield: Optimal, efficient reconstruction of Root-Unknown phylogenetic networks with constrained and structured recombination. *JCSS*, 70, 381-398, 2005.
9. D. Gusfield, S. Eddhu and C. Langley: Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinformatics and Computational Biology*, v. 2, 173-213, 2004.
10. D. Gusfield, S. Eddhu and C. Langley: The fine structure of galls in phylogenetic networks. *INFORMS J. on Computing*, v. 16, p.459-469, 2004.
11. J. Hein: Reconstructing evolution of sequences subject to recombination using parsimony. *Math. Biosci.*, v. 98, p.185-200, 1990.
12. J. Hein, A heuristic method to reconstruct the history of sequences subject to recombination. *J. Mol. Evol.*, v.36, p.396-405, 1993.
13. J. Hein, M. Schierup and C. Wiuf: *Gene Genealogies, Variation and Evolution: A primer in coalescent theory*. Oxford University Press, 2005.
14. D. Hinds, L. Stuve, G. Nilsen, E. Halperin, E. Eskin, D. Gallinger, K. Frazer and D. Cox: Whole-Genome Patterns of Common DNA variation in three human populations. *Science*, v. 307, p.1072-1079, 2005.

15. R. Hudson and N. Kaplan: Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, v.111, p.147-164, 1985.
16. International HapMap Consortium: The HapMap project, *Nature*, 426, p789-796, 2003.
17. International HapMap Consortium: A haplotype map of the human genome. *Nature*, 437, p1299-1320, 2005.
18. B. Kerem, J. M. Rommens, J. A. Buchanan, D. Markiewicz, T. K. Cox, A. Chakravarti, M. Buchwald and L. C. Tsui: Identification of the cystic fibrosis gene: genetic analysis. *Science*, 245, p1073-1080, 1989.
19. F. Larribe, S. Lessard and N. J. Schork: Gene mapping via Ancestral Recombination Graph. *Theor. Popul. Biol.*, v. 62, p. 215-229, 2002.
20. N. Li and M. Stephens: Modeling Linkage Disequilibrium, and identifying recombination hotspots using SNP data. *Genetics*, 165, p2213-2233, 2003.
21. R. Lyngso, Y. S. Song and J. Hein: Minimum Recombination Histories by Branch and Bound. *Proceedings of Workshop on Algorithm of Bioinformatics (WABI) 2005*, v.3692, p.239-250.
22. M. S. McPeck and A. Strahs: Assessment of linkage disequilibrium by the decay of haplotype sharing, with application to fine-scale genetic mapping. *Am. J. Hum. Genet.*, 65, p858-875, 1999.
23. M. Minichiello and R. Durbin: Mapping trait loci using inferred ancestral recombination graphs. *Am. J. Hum. Genet.*, v.79, p.910-922, 2006.
24. A. P. Morris, J. C. Whittaker and D. J. Balding: Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. *Am. J. Hum. Genet.*, 70, p686-707, 2002.
25. S. R. Myers and R. C. Griffiths: Bounds on the minimum number of recombination events in a sample history. *Genetics*, v. 163, p.375-394, 2003.
26. M. Norborg and S. Tavaré: Linkage disequilibrium: what history has to tell us. *Trends in Genetics*, 18, 83-90, 2002.
27. B. Rannala and J. P. Reeve: High-resolution multipoint linkage-disequilibrium mapping in the context of a human genome sequence. *Am. J. Hum. Genet.*, 69, p159-178, 2001.
28. N. Risch and K. Merikangas: The Future of Genetic Studies of Complex Human Diseases. *Science*, 275, p1516-1517, 1996.
29. Y. Song and J. Hein: Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events. *Proc. of 2003 Workshop on Algorithms in Bioinformatics (WABI)*, 2003.
30. Y. Song and J. Hein: On the Minimum Number of Recombination Events in the Evolutionary History of DNA Sequences. *J. of Math. Biology*, v.48, p.160-186, 2003.
31. Y. S. Song, Y. Wu and D. Gusfield: Efficient computation of close lower and upper bounds on the minimum number of needed recombinations in the evolution of biological sequences. *Bioinformatics*, v. 421, p.i413-i422, *Proceedings of ISMB 2005*.
32. M. Stephens, N. Smith and P. Donnelly: A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.*, v. 68, p.978-989, 2001.
33. A. R. Templeton, E. Boerwinkle and C. F. Sing: A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping. I. Basic theory and an analysis of alcohol dehydrogenase activity in *Drosophila*. *Genetics*, 117, p343-351, 1987.
34. L. Wang, K. Zhang and L. Zhang: Perfect Phylogenetic Networks with Recombination. *J. of Comp. Bio.*, v.8, p.69-78, 2001.
35. S. Zollner and J.K. Pritchard: Coalescent-Based Association Mapping and Fine Mapping of Complex Trait Loci. *Genetics*, 169, p. 1071-1092, 2005.