Pacific Symposium on Biocomputing 15:348-358(2010)

# DETECTING GENOME-WIDE HAPLOTYPE POLYMORPHISM BY COMBINED USE OF MENDELIAN CONSTRAINTS AND LOCAL POPULATION STRUCTURE

XIN LI

YIXUAN CHEN

JING LI*

*Department of Electrical Engineering and Computer Science*
*Case Western Reserve University*
*Cleveland, OH 44106, United States*
*E-mail: {xin.li2,yixuan.chen,jingli}@case.edu*

Data from current gene-disease association studies motivate changes to existing haplotype inference methodologies. Many datasets are now comprised of both pedigree and population data so it is desirable to incorporate both sources of information when inferring haplotypes. The availability of high-density SNP data also makes it possible to determine and use the precise locations of recombination events. Our proposed method reconstructs haplotype structure on a genome-wide level by jointly using the information from the Mendelian law of inheritance and local population structure. The method combines in one framework new techniques of recombination event detection, maximum likelihood optimization of population haplotype diversity and our previous algorithm of zero-recombinant haplotype reconstruction. Experiments on both real and simulated datasets prove the efficiency and accuracy of our approach in reconstructing the haplotype structure. Our method makes it possible to reveal the haplotypic variation on a genome-wide level.

*Keywords*: haplotypic variation, genetic linkage, Mendelian law

## 1. Introduction

Haplotypes are mostly obtained by computational methods from genotype data instead of directly by molecular methods due to the high cost of the current technology. Haplotypes if corrected inferred provide exact information on the linkage of SNPs, which is of substantial importance in detecting gene-disease association.[1,4,15] Typically haplotype inference from pedigree and population data is performed separately. Methods on pedigree data[12,18] make use of the Mendelian law of inheritance and some parsimony criteria on the number of recombination events. Due to the enrichment of constraints in pedigrees, such methods are usually fast. Methods on population data such as fastPHASE[16] and Beagle[5] make use of the clustering property of haplotypes in the population over short regions. Due to the enumerative nature of these methods, they are usually slow. With respect to haplotyping accuracy, previous analyses[13] have demonstrated that using family constraints alone achieves better accuracy than using population information alone. On the other hand, population information works better than family constraints in imputing missing genotypes. Therefore, it is most desirable to jointly use the family and population information. Beagle recently added the function to process data sets of many trios and parent-offspring pairs.[6] The authors reported that Beagle achieved extreme accurate results on data sets of trios compared to data sets of unrelated individuals due to the use of the rules of Mendelian inheritance.

Linkage analysis[7,9] can actually yield the maximum likelihood haplotype configuration in terms of both family constraints and population haplotype frequency by enumerating every possible inheritance pattern and every possible allele assignment. However the time complexity of this approach is exponential to either the number of markers or the pedigree size, thus it is infeasible for any reasonably large dataset. To avoid exhaustive enumeration, it is critical to represent the set of all compatible family configurations in a compact form. Li and Li[14] found that by assuming no recombination it is possible to represent the set of family configurations as a linear span of variables that can be found by solving a linear system of binary variables representing inheritance and phase. However, to apply this method to whole-genome data, we must first identify recombination positions in each family such that we can segment the chromosomes into recombinant-free regions.

*Corresponding author.

Pacific Symposium on Biocomputing 15:348-358(2010)

Furthermore, to find an optimal solution using population haplotype frequency, we need a computationally feasible way to search the solution space since the number of possible haplotypes is potentially exponential to the number of markers. In this paper, we present a framework combining the efforts of recombination detection, zero-recombinant haplotype inference, and local haplotype structure clustering to jointly use the family and population information. Our method makes it possible to accurately reveal the haplotype structure in human populations on a genome-wide level. The algorithm is implemented in *C++* and is freely available upon request.

## 2. Methods

The overall flow of the method **MML** (**M**endalian Constrained **M**aximum **L**ikelihood) is staged in three steps as illustrated in Algorithm 2.1.

**Algorithm 2.1** MML

(1) Infer recombination positions for each family and each chromosome. Partition the chromosomes according to recombination positions.
(2) On each pedigree, for each of the zero-recombinant segments, apply DSS[14] (our previously developed algorithm to handle Mendelian constraints) to establish the solution space under Mendelian and zero-recombinant constraints.
(3) Search the solution space (obtained in (2)) for the optimal solution with maximum likelihood based on population haplotype frequency.

In step 1, we infer recombination positions in each nuclear family of the pedigree by analyzing identical by descent (IBD) status of alleles between each sibling pair. Based on the inferred recombination positions, we partition chromosomes into segments such that every segment is recombinant-free. In step 2, we derive all possible configurations of a pedigree under Mendelian and zero-recombinant constraints for each recombinant-free segment obtained in step 1. This is done by using our previous algorithm DSS.[14] DSS can output a compact description of all compatible solutions as a linear space. In step 3, we use haplotype frequencies in the population to identify the optimal haplotype configuration of each pedigree. We will describe step 1, step 2 and step 3 in Sec. 2.1, Sec. 2.2 and Sec. 2.3 respectively.

### 2.1. *Detect Recombination Events in Families with Dense Markers*

Recombination events are implied if a common inheritance vector for a segment of loci that satisfies Mendelian constraints cannot be found. Typically, there is uncertainty as to how many recombination events occur and at which loci or in which individuals these events occur. Usually, such parsimony criteria as minimum number of recombinants[12] are used to find a possible assignment. However, with the availability of densely marked data, we can almost always fix the inheritance vector within each zero-recombinant region due to the enrichment of Mendelian constraints. Consequently, we can also develop special techniques to localize the recombination positions with minimal ambiguity.

For each nuclear family, we look at the IBD status of the alleles and its sibling pairs to detect a recombination position. The change of IBD status from one locus to another indicates a change in the inheritance pattern, that is, a recombinant. Loci of a father (similarly for a mother) can be divided into three categories depending on their informativeness in determining the paternal IBD status of a sibling pair.

(1) informative: he is heterozygous, and the phases of both children are determined at this locus.
(2) semi-informative: he is heterozygous, and at least one of the children is not phased at this locus.
(3) non-informative: he is homozygous at this locus.

Pacific Symposium on Biocomputing 15:348-358(2010)

In situation (1), since the father is heterozygous, the IBD status and the IBS (identical by state) status of the paternal alleles of a sibling pair are equivalent. In situation (2), the IBD status of the paternal alleles is not determined, but it is dependent on the IBD status of the maternal alleles. If we can somehow resolve the IBD status of the maternal alleles at this locus, we can also infer the IBD status of the paternal alleles. Note that in this situation, the mother must be heterozygous, otherwise all children would be phased. In situation (3), this locus provides no information about the paternal IBD status of any of the sibling pairs.

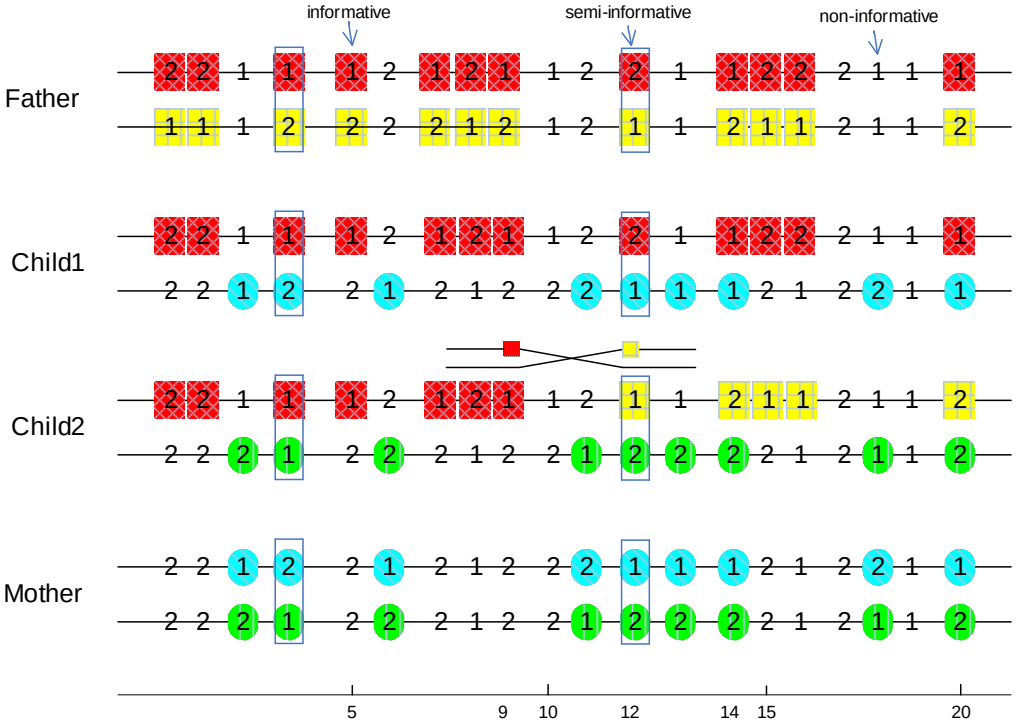



Fig. 1. A segment of a chromosome from a nuclear family with 2 children. Colored nodes are informative alleles which have determined IBD status between these two siblings, with squares and circles representing paternal and maternal alleles respectively. Remaining alleles are non-informative. At each locus, alleles colored the same are identical by descent(IBD). A frame around a pair of alleles indicates that they are semi-informative, and this pair of alleles infers IBD between two siblings if in the context of the informative loci nearby. From the IBD status between siblings, we can infer a paternal recombination event between the 9th and 12th locus, but with the ambiguity whether it happens in Child1 or Child2. We suppose the recombinant is in Child2 for illustration purposes.

Informative loci give a narrow-spaced probing on the IBD status of the whole chromosome. We can detect recombination events by observing the change of IBD status of alleles among these informative loci. By doing so, however, we may miss possible double recombination events that do not manifest a change in the IBD status between two nearby informative loci. If we assume markers are dense, however, the possibility of a double recombination event within a short distance is negligible. Fig. 1 shows an example on how to detect recombination positions in a nuclear family. By using informative markers, we could infer a paternal recombination event between the 9th and 14th locus.

Semi-informative loci can help further localize the recombination position because it is almost impossible for a paternal and a maternal recombination event to occur coincidentally within a short region. If a semi-informative locus falls between two informative loci indicating different paternal IBD status, we can assume no recombination on the maternal side and let its maternal IBD status follow that of its surrounding informative loci. By assuming the maternal IBD status, we can now infer the paternal IBD status for this semi-informative locus. For example, in Fig. 1, at the 12th loci, by assuming that Child1 and Child2 are not IBD for their

Pacific Symposium on Biocomputing 15:348-358(2010)

maternal alleles, we infer that the sibling pair are also not IBD for their paternal alleles, so that we could refine the recombination position to be between the 9th and 12th locus.

Since ambiguous intervals of recombination events now only contain non-informative markers which are compatible with any inheritance pattern, we may pick any position within such intervals to partition a chromosome into recombinant-free segments. Notice that for non-informative loci, the phases of all family members are actually fixed, and thus the choice of a recombination position will not influence the final haplotype configuration.

To determine the individual in which the recombination event actually occurs, we can look at the IBD status of all sibling pairs. If we observe that the IBD status changes between a specific child $i$ and any of the other children, while there is no change among these children themselves, then child $i$ carries the recombinant. However, if the nuclear family has only two children, then the ambiguity is unresolvable in this way. Notice that the assignment of recombination to a different child will result in a different haplotype configuration in the parent. Therefore in this situation, we can use population haplotype frequency to suggest a most probable assignment.

### 2.2. *Establish the Solution Space under Mendelian Constraints*

We can use two types of binary variables: $p$ variables and $h$ variables, to encode the Mendelian constraints in a pedigree. A $p$ variable indicates the phase of two alleles of an individual. $p^a = 0$ means that the smaller-numbered allele ("1") of individual "$a$" is of paternal source, $p^a = 1$ means it is of maternal source. An $h$ variable indicates the inheritance relationship between a parent-child pair, $h^{ab} = 0$ means that the paternal allele of individual "$a$" is transmitted to individual "$b$", $h^{ab} = 1$ means that the maternal allele is transmitted. Fig. 2 gives an example of the relationship between the $p$ variables and the $h$ variable of a parent-child pair under Mendelian constraints. As established in previous work,[14,18] Mendelian law can be expressed as linear equations of $h$ variables and $p$ variables.

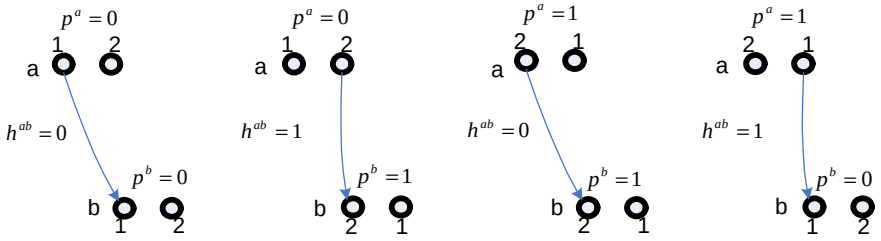



Fig. 2. Individual a is the father of individual b. In the situation where both a and b are heterozygous, relationship between $p$ variables and the $h$ variable can be expressed as $p^b = p^a + h^{ab}$.

If we assume no recombination within a certain number of loci, the $h$ variable between a parent-child pair, which indicates the inheritance patterns, should be the same for each of these loci. In this case, we can put constraints on $h$ variables from different loci together to form a single linear system. Li and Li[14] discover an almost linear time algorithm, DSS, to obtain a general solution to such a system. Here, a general solution means a description of all solutions as a linear span of variables.

The establishment of a general solution is important because it facilitates the search in the solution space for particular solutions to satisfy specific properties. The freedom in the solution space can be partitioned into two parts: the freedom of the inheritance vector (all $h$ variables) and the freedom of the allele assignment (all $p$ variables) under a fixed inheritance vector. Experiments[14] have shown that the inheritance vector is usually fixed for a segment of 100 or more loci. Once the inheritance is determined, the relationship between alleles of different individuals is determined with only 1 degree of freedom (if all members of the pedigree are heterozygous) or no degrees of freedom (if one or more members are homozygous). Fig. 3(a) shows an example

Pacific Symposium on Biocomputing 15:348-358(2010)

for the first situation. In the case of a missing genotype, there might be an increase in degrees of freedom (Fig. 3(b)). By applying the Mendelian and zero-recombinant constraints, we can greatly reduce the search space for finding the maximum likelihood solution using population local structure, which will be discussed in Sec. 2.3.

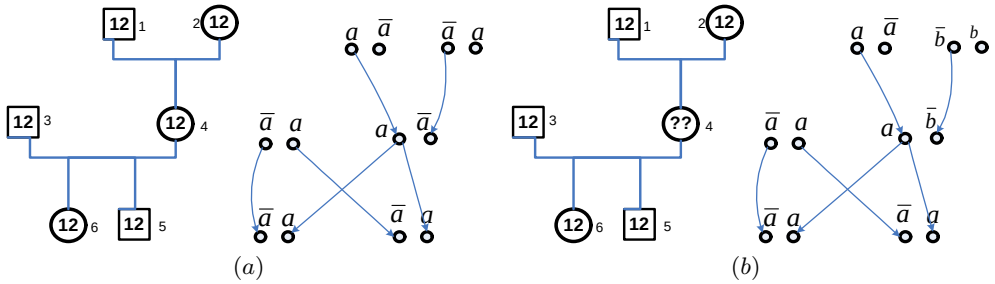



Fig. 3. (a) A pedigree of 6 individuals. All individuals are heterozygous with genotype "12" at a certain locus. For this fixed inheritance vector, the relationship between alleles of different individuals is determined. $a$ is a variable denoting the status of an allele and $\bar{a}$ is its complementary status. (b) Same pedigree and inheritance vector as (a), but at a different locus with genotype missing at individual 4. In this case, there are two degrees of freedom represented by variables $a$ and $b$.

### 2.3. *Maximum Likelihood Solution Based on Population Haplotype Frequency*

The inheritance vector ($h$ variables between each parent-child pair) specifies how founder haplotypes are transmitted to every descendant of a pedigree and the configuration of a pedigree is fully determined by the inheritance vector and the founder haplotypes. If the inheritance vector is fixed, the likelihood of a configuration of a pedigree is simply the product of founder haplotype probabilities.

In Sec. 2.2, we describe how Mendelian and zero-recombinant constraints provide a small candidate set of all possible haplotype configurations for each pedigree. Next, we need to pick a solution of maximum likelihood from this candidate set based on haplotype frequencies. Since the actual haplotype frequencies in the population are unknown, we use an EM (Expectation Maximization) procedure to find the optimal solution. The procedure is described in Algorithm 2.2. The initial pool of haplotype frequencies is generated by randomly sampling from founder haplotypes within the solution space of each pedigree. In step (2), we search the solution space for an optimal configuration with the highest likelihood. In step (3), we update the haplotype frequency pool only with the optimal solution of each pedigree. It is different from conventional EM methods, where all possible solutions are updated into the pool weighted by their current likelihood. By adopting such an approximation, we can significantly hasten the optimization process by not traversing the entire solution space.

**Algorithm 2.2** Haplotype Frequency EM

(1) Build the initial pool of haplotype frequencies by randomly sampling from the solution space of each pedigree.
**repeat**
(2) Find the optimal solution with maximum likelihood based on the current pool.
(3) Update the pool with the haplotype frequencies of optimal solutions obtained in (2).
**until** convergence is achieved

#### 2.3.1. *Probabilistic prefix tree for fast branch-and-bound optimization*

We create a data structure called "probabilistic prefix tree" to facilitate the search of the optimal configuration in the solution space. A probabilistic prefix tree is essentially a binary search tree which encodes the frequencies

Pacific Symposium on Biocomputing 15:348-358(2010)

of each haplotype and their prefixes. It provides quick indexing for haplotype frequencies and can be updated dynamically using conventional binary search tree techniques. Each leaf node in the tree represents a haplotype and each internal node represents a prefix. The frequencies of internal nodes can be generated by simply summing up the frequencies of all leaf nodes of its subtree. Fig. 4 shows an example of a probabilistic prefix tree.

As discussed in Sec. 2.2, for a fixed inheritance vector, the relationship between alleles of different family members is fixed at each locus. On a pedigree of $n$ founders and $m$ markers, we do a depth first search from locus 1 to locus $m$ of a haplotype, where for each locus we pick an assignment for all $2n$ founder alleles if there is one or more degrees of freedom. Meanwhile, we calculate the likelihood of the pedigree up to the current locus which is the product of frequencies of the founder haplotype prefixes ending at locus $i$: $\prod_{j=1..2n} freq(h_i^j)$. Since the frequency of any haplotype prefix is greater than that of the entire haplotype, if the likelihood drops below the bound, we backtrack for there is no possibility of a better solution. Otherwise, we move to the next locus until we reach $m$. If we achieve a higher likelihood, we replace the bound with the new likelihood and record the current best configuration.

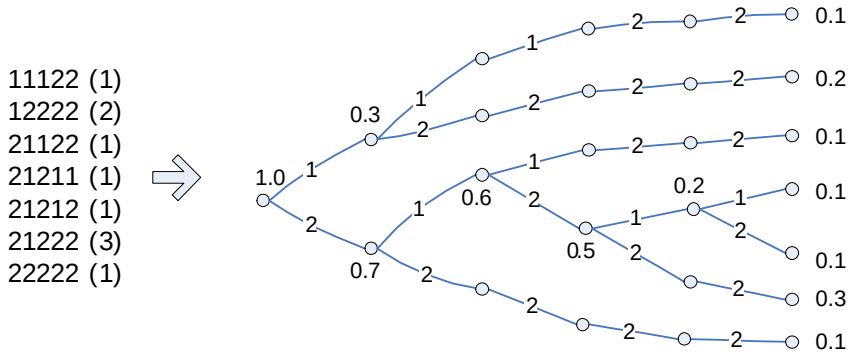



Fig. 4. On the left is the haplotypes and their count (in brackets) in the population. On the right is the probabilistic prefix tree after adding all these haplotype to an empty tree. Some nodes are annotated with the normalized frequencies of the corresponding haplotypes or haplotype prefixes.

## 3. Experimental Results

### 3.1. *Detect Recombination Events and Haplotype Diversity*

We use MML to analyze haplotype polymorphisms in a real human population. There are 32250 markers spanning a region of 170 million base pairs on chromosome 6, with an average marker interval distance of 5kb. Missing genotype rate is 0.12% and typing error rate (as reflected by Mendelian inconsistency) is 0.11%. There are 3 isolated individuals and a total of 193 nuclear families, among which 112 have 2 children and 81 have 1 child.

From 112 families with 2 children, we infer 322 paternal and 535 maternal recombination events. Fig. 6 shows the resolution of the inferred recombination positions. 82% of the recombination events can be localized within an interval less than 100kb, and 53% within an interval less than 30kb.

Fig. 5(c) shows the averaged degree of freedom at each locus of a family after applying the Mendelian and zero-recombinant constraints. Based on the actual heterozygosity rate of the current dataset, there is expected to be 1.3347 degrees of freedom on a family of 2 children or 0.9982 degrees of freedom on a family of 1 child at each locus. By exploiting these constraints first, we have eliminated more than 95% of the phasing freedom of a family. A big family size will result in fewer degrees of freedom due to the increased number of constraints.

As shown in Fig. 5(a), the haplotype diversity varies for different locations of the chromosome. In the initial sampling (Fig. 5(b)), 23.41%(8.41%) of the most common haplotypes covers 90%(80%) of the total frequency. This indicates that most of the common haplotypes are recovered and sampled multiple times.

Pacific Symposium on Biocomputing 15:348-358(2010)

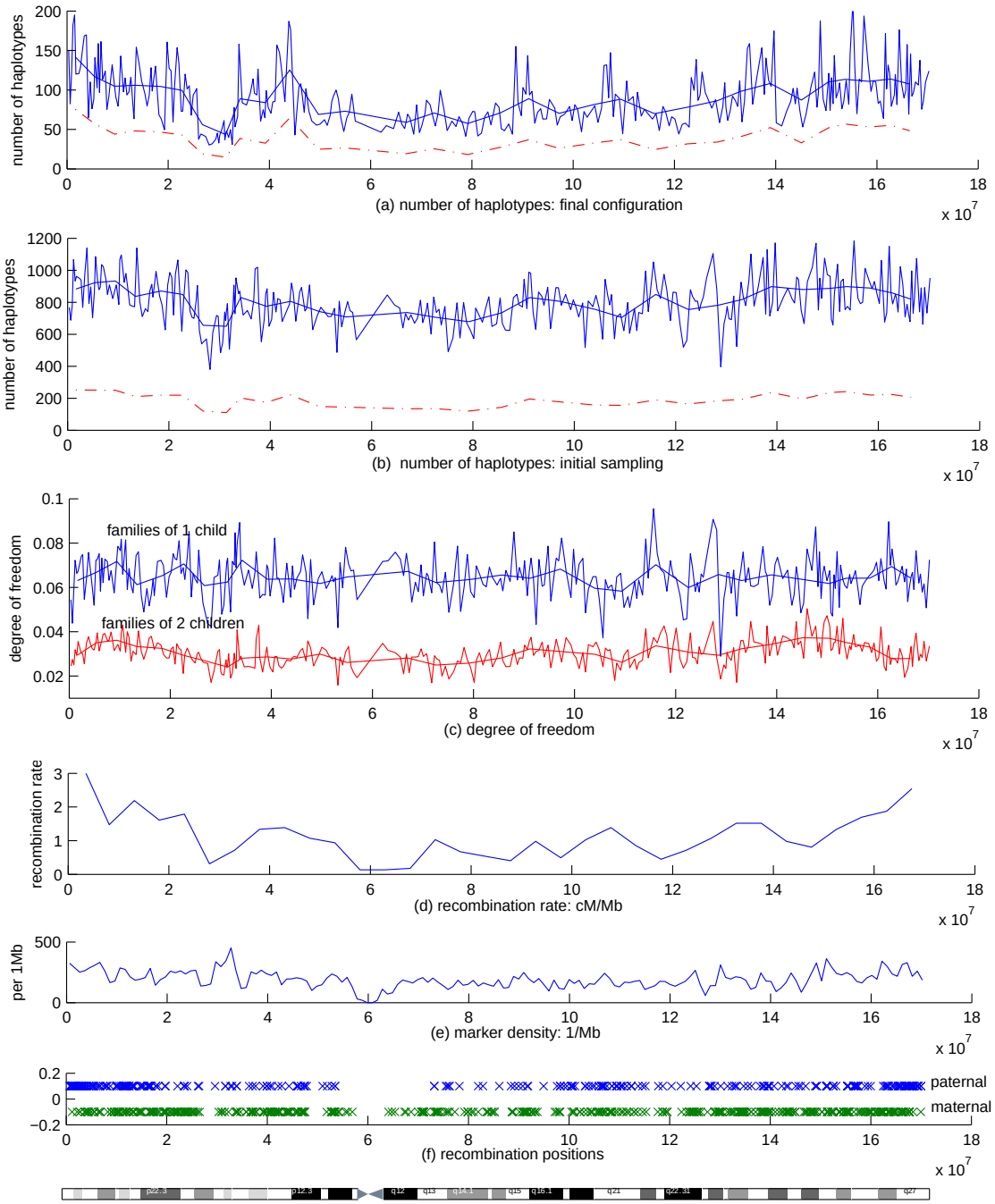



Fig. 5. X axis is the location as in base pairs on chromosome 6. (a)(b) Two charts show the number of haplotypes within each segment of 20 markers across chromosome 6, in the final configuration and the initial sampling respectively. Two solid lines are average numbers smoothed over 5 and 50 segments. The dashed line is the number of the most common haplotypes covering 90% of the total frequency. (c) Degree of freedom at each locus under Mendelian and zero-recombinant constraints. The lines are averaged values over all pedigrees of 1 child (upper) and 2 children (lower) respectively. Results are smoothed over 100 and 1000 markers. (d) Recombination rate in terms of centimorgan per million base pairs. (e)(f) The bottom two charts show the marker density, the paternal and maternal recombination positions over the whole chromosome. There are no markers around the centromere region.

Pacific Symposium on Biocomputing 15:348-358(2010)

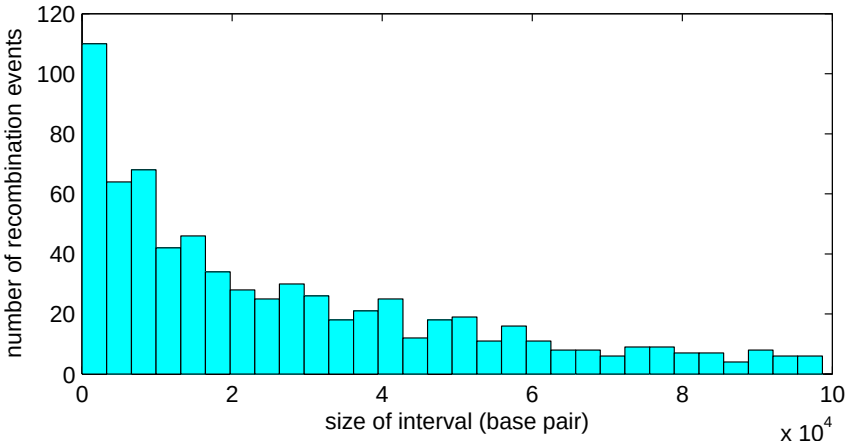



Fig. 6. The distribution of the length of ambiguous intervals of inferred recombination positions.

### 3.2. *Evaluation of Accuracy and Scalability*

We used the Cystic Fibrosis Transmembrane-Conductance Regulator (CFTR) Gene Data Set[8] for small scale testing and Simulated Rheumatoid Arthritis (RA) Data from Genetic Analysis Workshop (GAW) 15 for genome-wide testing of MML. We also compare the performance of our program with the conventional linkage analysis approach. Here, we use the statistical tool package Merlin.[2] Merlin can be used to perform haplotype inference on datasets of family and population mixed type. It first evaluates the inheritance vector in each family by exhaustive enumeration. Then it uses an EM approach to obtain the maximum likelihood configuration based on the population haplotype frequency. In order to deal with large numbers of markers, Merlin groups the markers by some pre-determined length and generates one single inheritance vector for each segment by assuming no recombination. However, if there does exist recombination in a segment, the program will fail. In Sec. 3.2.1, we compare MML and Merlin on small lengths of markers with different pedigree sizes to examine the efficiency by explicitly using Mendelian constraints rather than pure enumeration. On a genome-wide level (Sec. 3.2.2), our program can still successfully reveal the actual haplotype structure when Merlin is not applicable due to unresolved recombination.

#### 3.2.1. *Influence of pedigree size, missing rate on performance*

We simulate pedigrees with no recombination to evaluate the performance of MML on zero-recombinant segments with different data settings. Pedigrees are generated with different sizes (4, 17, 29, 52) and missing rates (0.00, 0.05, 0.10, 0.15, 0.20) by using *SimPed*.[10] We pick from the CFTR data a subset of 29 distinct haplotypes of 19 markers spanning a region of 1.8Mb on chromosome 7q31. *SimPed* will assign founder haplotypes by sampling from the given set and transmit them onto other family members assuming no recombination. Each population has 500 families of a given parameter setting, and we average our results over 10 replicates of a population. The accuracy and running time comparison between MML and Merlin are shown in Fig. 7.

By explicitly exploiting the Mendelian constraints instead of enumerating all possible inheritance vectors, MML can achieve much greater time efficiency than Merlin on large pedigrees or on high missing rates (Fig. 7(b)). Both methods achieve better accuracy with large pedigrees (Fig. 7(a)) due to increased family constraints and population information. By adopting an approximate EM algorithm instead of traversing the whole search space, MML is of negligible accuracy difference to Merlin. This demonstrates the approximation approach to be a reasonable trade-off for efficiency. This is further confirmed on a large pedigree size of 52 (table in Fig. 7), where the performance of Merlin crashes with a high error rate (up to 30%) and exponentially longer running time due to too much freedom in resolving the inheritance vector. On the other hand, MML exhibits very robust consistency in both accuracy and efficiency.

Pacific Symposium on Biocomputing 15:348-358(2010)

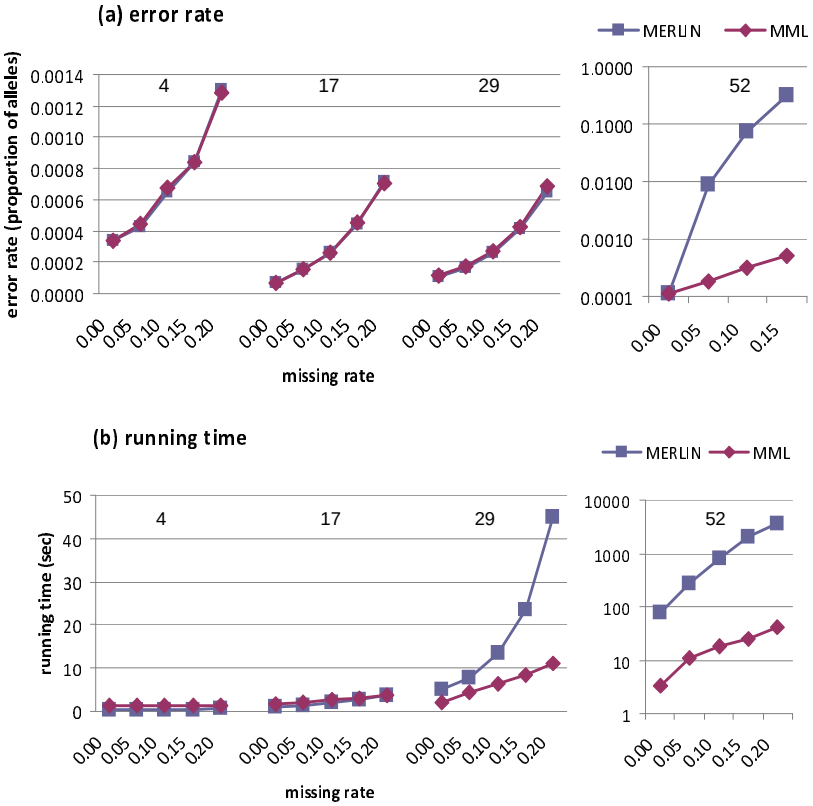
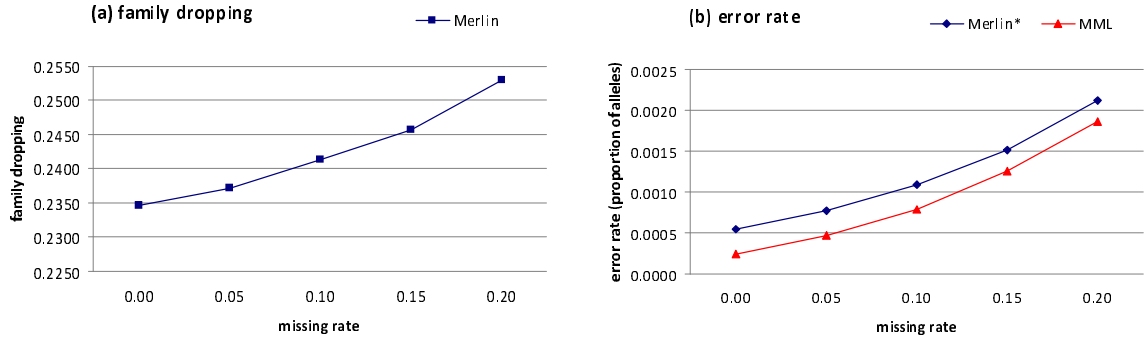



Fig. 7. Comparison of two methods on a dataset of 500 pedigrees. Performance is examined on 4 different pedigree sizes: 4, 17, 29, 52 and 5 different missing genotype rates: 0.00, 0.05, 0.10, 0.15, 0.20. Error rate is calculated by comparing the allele-by-allele difference between the inferred the haplotype and the correct haplotype.

### 3.2.2. *Genome-wide haplotype inference accuracy*

We tested MML and Merlin on chromosome 6 of the RA data which has 17820 SNPs with an average inter-marker spacing of 9586bp. The RA data consisted of 100 replicates, each with 1500 nuclear families (two parents and two offsprings). We used 500 out of the 1500 families and averaged our results over 10 replicates. We artificially set up to 20% genotype to missing to estimate the robustness of the methods against missing data.




(a) (b)

Fig. 8. Performance of MML and Merlin on chromosome 6 of RA data. Missing rates are 0.00, 0.05, 0.10, 0.15, 0.20. *The error rate of Merlin is based only on the families it successfully processes.

Pacific Symposium on Biocomputing 15:348-358(2010)

As shown in Fig. 8(b), MML can successfully reconstruct the haplotypes of each individual with an allele-by-allele error rate of 0.025% (for a missing rate of 0%) to an error rate of 0.19% (for a missing rate of 20%). Since Merlin assumes no recombination within each pre-defined segment, it fails on 25% of the 500 families (Fig. 8(a)). On the remaining families, it happens that all recombination events have ambiguous intervals riding across segment boundaries instead of contained completely in a single segment such that Merlin can still find a single inheritance vector for each segment. However, the overall accuracy of MML on all families is even better than the accuracy of Merlin on these retained families (Fig. 8(b)).

## 4. Conclusions

In order to reveal the human haplotypic variation on a genome-wide level, we need to overcome the computational difficulties complicated by huge numbers of markers, large pedigree and population size, and substantial numbers of missing genotypes. We applied our previous algorithm to find and compactly represent the subset of pedigree inheritance configurations that are consistent with the Mendelian law. We develop new techniques to resolve recombination positions in densely marked sequences, and a quick search strategy for detecting haplotype combinations of maximum likelihood in a population. All these techniques make it possible to handle large degrees of freedom in the pedigree inheritance patterns, the uncertainty of recombination positions, and the variety of possible haplotypes of a population. The combined exploitation of Mendelian constraints and local population structure makes the most of the current data designs to restore the underlying haplotype polymorphism. Experimental results on both real and simulated populations show that our method can successfully reconstruct the haplotypes with high accuracy and it is scalable in terms of both the pedigree (population) size and the missing genotype rate.

## Acknowledgments

We would like to thank Dr. Fengyu Zhang and Matthew Hayes for helpful discussions. This research is supported by National Institutes of Health/National Library of Medicine grant LM008991, and in part by National Institutes of Health/National Center for Research Resources grant RR03655. Support for generation of the GAW15 simulated data was provided from NIH grants 5RO1-HL049609-14, 1R01-AG021917-01A1, the University of Minnesota, and the Minnesota Supercomputing Institute. We would also like to acknowledge GAW grant R01 GM031575.

## References

1. The International HapMap Consortium, A second generation human haplotype map of over 3.1 million SNPs, *Nature* **449**:851–61, 2007.
2. Abecasis GR, Cherny SS, Cookson WO, Garden LR, Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. *Nature Genetics* **30(1)**:97–101, 2002.
3. Abecasis GR, Wigginton JE, Handling marker-marker linkage disequilibrium pedigree analysis with clustered markers, *American Journal of Human Genetics* **77**:754–767, 2005.
4. Bader JS, The relative power of SNPs and haplotype as genetic markers for association tests, *Pharmacogenomics* **2(1)**:11–24, 2001.
5. Browning SR, Browning BL, Rapid and accurate haplotype phasing and missing data inference for whole genome association studies using localized haplotype clustering, *American Journal of Human Genetcis* **81**:1084-1097, 2007.
6. Browning BL, Browning SR, A unified approach to genotype imputation and haplotype phase inference for large data sets of trios and unrelated individuals, *American Journal of Human Genetcis* **84**:210-223, 2009.
7. Elston RC, Stewart J, A general model for the genetic analysis of pedigree data, *Human Heredity* **21**:523–542, 1971.
8. Kerem B, Rommens JM, Buchanan JA, Markiewicz D, Cox TK, Chakravarti A, Buchwald M, et al, Identification of the cystic fibrosis gene: genetic analysis, *Science* **245**:1073-1080, 1989.
9. Lander ES, Green P, Construction of multilocus genetic linkage maps in humans, *The Proceedings of the National Academy of Sciences* **84**:2363–2367, 1987.
10. Leal SM, Yan K, Müller-Myhsok B, SimPed: a simulation program to generate haplotype and genotype data for pedigree structures, *Human Heredity* **60**:119–122, 2005.

Pacific Symposium on Biocomputing 15:348-358(2010)


11. Li J, Jiang T, A survey on haplotyping algorithms for tightly linked markers, *Journal of Bioinformatics and Computational Biology* **6(1)**:241–259, 2008.
12. Li J, Jiang T, Computing the minimum recombinant haplotype configuration from incomplete genotype data on a pedigree by integer linear programming, *Journal of Computational Biology* **12**:719–739, 2005.
13. Li X, Li J, Comparisons of haplotype Inference from pedigree data and population data, *BMC Proceedings* **1**:S55, 2007.
14. Li X, Li J, An almost linear time algorithm for a general haplotype solution on tree pedigrees with no recombination and its extensions, Journal of Bioinformatics and Computational Biology **7(3)**:521-545, 2009.
15. Morris RW, Kaplan NL, On the advantage of haplotype analysis in the presence of multiple disease susceptibility alleles, *Genetic Epidemiology* **23**:221–233, 2002.
16. Scheet P, Stephens M, Fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase, *American Journal of Human Genetics* **78**:629–644, 2006.
17. Sobel E, Lange K, Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics, *American Journal of Human Genetics* **58(6)**:1323–1337, 1996.
18. Xiao J, Liu L, Xia L, Jiang T, Fast elimination of redundant linear equations and reconstruction of recombination-free mendelian inheritance on a pedigree, *Proc. of 18th Annual ACM-SIAM Symoposium on Discrete Algorithms (SODA'07)*, 655–664, 2007.