k-Recombination Haplotype Inference in Pedigrees

Francis Y.L. Chin[1], Qiangfeng Zhang[2], and Hong Shen[3]

[1] Department of Computer Science,
The University of Hong Kong, Pokfulam, Hong Kong
chin@cs.hku.hk
[2] Department of Computer Science,
University of Science and Technology of China, Hefei, China
qfzhang@mail.ustc.edu.cn
[3] Graduate School of Information Science, JAIST, Ishikawa, Japan
shen@jaist.ac.jp

Abstract. Haplotyping under the Mendelian law of inheritance on pedigree genotype data is studied. Because genetic recombinations are rare, research has focused on Minimum Recombination Haplotype Inference (MRHI), i.e. finding the haplotype configuration consistent with the genotype data having the minimum number of recombinations. We focus here on the more realistic k-MRHI, which has the additional constraint that the number of recombinations on each parent-offspring pair is at most k.

Although k-MRHI is NP-hard even for k = 1, we give an algorithm to solve k-MRHI efficiently by dynamic programming in O(nm03k+12m0) time on pedigrees with n nodes and at most m0 heterozygous loci in each node. Experiments on real and simulated data show that, in most cases, our algorithm gives the same haplotyping results but runs much faster than other popular algorithms.

1 Introduction

The modeling of human genetic variation is critical to the understanding of the genetic basis for complex diseases. *Single nucleotide polymorphisms* (*SNPs* [13]) are the most frequent form of variation. The Human Genome Project and other large-scale efforts have identified millions of SNP markers. Although each marker can be analyzed independently, it is more informative to analyze them in groups. Therefore, it is useful to analyze *haplotypes* (*hap*loid geno*types*), which are sequences of linked markers on a single chromosome. In diploid organisms, such as humans, chromosomes come in pairs, and experiments often yield *genotype* information, which blends haplotype information for chromosome pairs. Given that deriving haplotype information experimentally is time-consuming and expensive, there is increasing interest in inferring haplotype information, or *haplotyping*, computationally [4][7] from genotype information.

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 985 – 993, 2005.

Haplotyping *pedigree* data is believed to be more reliable than haplotyping population data for unrelated individuals, and genetic research shows that recombinations are rare in human pedigree data [6]. The *Minimum-Recombination Haplotype Inference* (*MRHI*) problem, which is NP-hard [5], is to find a haplotype configuration with minimum number of recombinations given pedigree genotype data [8][9][12][15]. At times, however, the MRHI model might yield unrealistic results in which recombinations are concentrated in a few parent-offspring pairs. The *k*-MRHI problem is basically MRHI but with the additional constraint that the number of recombinations in each parent-offspring pair is bounded by a constant *k*.

The *k*-MRHI problem is NP-hard even for $k = 1$, but can be solved by a dynamic programming (DP) algorithm similar to the algorithm in [5] for MRHI. By avoiding studying all 2^{3m_0} haplotype configurations in each parents-offspring trio, our algorithm takes $O(nm_0{}^4 2^{m_0})$ time when $k = 1$, instead of $O(nm_0 2^{3m_0})$ [5] for the MRHI problem on pedigrees with n nodes and at most m_0 heterozygous loci in each node. Not all nodes have m_0 heterozygous loci, and the number of feasible haplotype configurations at a node is limited by that of its neighbors and thus can be much less than 2^{m_0}. This observation leads to the idea of choosing a good root node to speed up the algorithm.

This paper's main contributions are: (1) to define a more realistic problem for haplotype inference (*k*-MRHI), (2) to give a more efficient and practical DP algorithm for *k*-MRHI, and (3) to present an efficient algorithm to find a good root in the pedigree to improve the DP algorithm's performance.

2 Preliminaries

Haplotypes and genotypes consist of linked *genetic markers* which are small DNA segments. The physical position of a marker on a chromosome is called a *locus* and its state is called an *allele*. The two alleles of a biallelic (2-state) SNP can be denoted by '0' and '1', and a haplotype h with m loci is presented as a string of length m over $\{0,1\}^m$, and a genotype g as a string over $\{0,1,2\}^m$. Haplotype pair $<h_1, h_2>$ is *consistent* with genotype g if (a) the two alleles of h_1 and h_2 are the same at the same locus (a *homozygous* site), for example '0' (respectively '1'), then the corresponding locus of g should also be '0' (respectively '1'); otherwise, (b) the two alleles of h_1 and h_2 are different, then the corresponding site of g should be '2' (a *heterozygous* site).

Figure 1(a) shows the pictorial representation of a pedigree with 13 nodes. A square represents a male node, a circle a female node, and a black dot a mating node. A **pedigree** can be formally defined as a weakly connected directed a cyclic graph $P = (V, E)$, where $V = M \cup F \cup N$, with M stands for the male nodes, F the female nodes, N the mating nodes, and $E = \{(u, v)|\ u \in M \cup F$ and $v \in N$, or alternatively $u \in N$ and $v \in M \cup F\}$.

Figure 1(b) shows the graph representation of the pedigree given in Figure 1(a). A *nuclear family* comprises a father, a mother, and their children (see subgraph in the dotted square) and can also be represented by a mating node which connects them together. A *parents-offspring trio*, or just *trio*, consists of two parents and one of their children, and a *parent-offspring pair* (*PO-pair*) refers to a father and his child or a mother and her child. In this paper, we assume that the pedigree never forms a cycle if the directions of edges are ignored (no *mating-loop*).

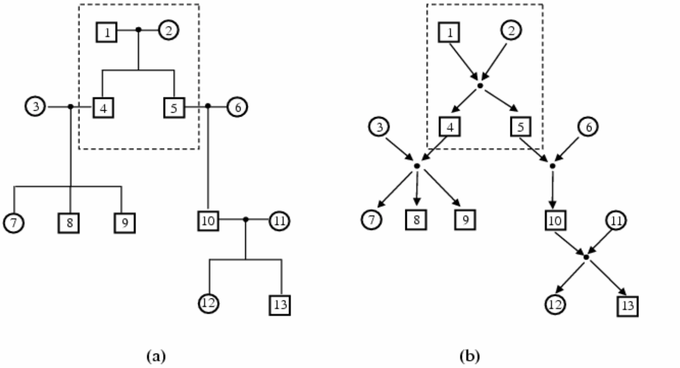

Fig. 1. The pictorial representation and graph representation of a pedigree

In the absence of genetic mutation, at each locus, the child must inherit one allele from its father and the other from its mother, i.e. the *Mendelian law of inheritance*. Usually, one haplotype of a child is inherited as a whole from one of the two haplotypes of a parent. However, *recombinations* may occur, where the two haplotypes of a parent get shuffled due to a crossover of a chromosome and one of the shuffled copies (*recombinant*) is passed on to the child. Genetic research shows that recombinations are rare in human genetics. The *Minimum Recombinant Haplotype Inference* (*MRHI*) problem [5] is to find the haplotype configuration such that the total number of recombinations in the whole pedigree is minimized.

Since it is rare to have many recombinations within one PO-pair, we focus on the *k-Recombination Haplotype Inference* (*k-MRHI*) problem to find a haplotype configuration that is consistent with the genotypes at all nodes having the minimum number of recombinations and no more than *k* recombinations in each PO-pair.

3 A Dynamic Programming Algorithm for *k*-MRHI

3.1 The 1-MRHI Problem ($k = 1$)

In [5], Doi *et al.* gives a proof for the NP-hardness of the MRHI problem, which trivially implies that the *k*-MRHI problem, even for $k = 1$, is also NP-hard. We shall focus on the 1-MRHI problem first and generalize to the *k*-MRHI problem later.

We adopt a locus-based dynamic programming (DP) approach to solve the 1-MRHI problem by assigning an arbitrary node R in the pedigree as the root and recursively finding $num[R][s]$, where $num[r][s]$ denotes the minimum number of recombinations required in the sub-tree rooted at r with the haplotype configuration s under the constraint that there is at most 1 recombination in each PO-pair of the sub-tree. If r has multiple mating nodes as its tree sons, we compute each mating node separately. Each child mating node of r, comprising father F, mother M and children C_1, ..., C_d. If r is a leaf node, $num[r][s] = 0$ for any haplotype configuration s; else, if r is M (or F, respectively) with haplotype configuration s, then:

$$num[r][s] = \min_{p}\{num[F][p] + \sum_{i=1}^{d} \min_{c_i}(num[C_i][c_i] + numtrio(p,s,c_i))\} \tag{1}$$

where p denotes the haplotype configuration at node F and c_i the haplotype configuration at C_i. $numtrio(p, s, c_i)$ returns the minimum number of recombinations required for a trio comprising F, M, and C_i with the haplotype configurations p, s and c_i respectively, under the constraint that no PO-pair can have more than one recombination. If there does not have any feasible solution, then $numtrio(p, s, c_i)$ returns ∞.

Similarly, if r is C_j with haplotype configuration s, then we have:

$$\begin{aligned} num[r][s] = \; & \min_{p,q}\{numtrio(p,q,s) + num[F][p] + num[M][q] \\ & + \sum_{i=1, i\neq j}^{d} \min_{c_i}(num[C_i][c_i] + numtrio(p,q,c_i))\}, \text{ where } r = C_j \end{aligned} \tag{2}$$

Note that the above algorithm is the same as that presented in [5], but a reduction in time complexity is possible because it is not necessary to consider all combinations of haplotype configurations in each trio, which number $O(2^{3m_0})$ in total. In fact, many combinations of haplotype configurations will be infeasible, i.e. have more than one recombination per PO-pair.

For example, assume the genotype of M is (2, 2, ..., 2) of length m_0 and with haplotype configuration $s = <h_{s1}, h_{s2}>$ and h_{c1} in the haplotype $c = <h_{c1}, h_{c2}>$ of C_i is inherited from s with no more than one recombination. There are m_0+1 ways of forming h_{c1} by inheriting its first w alleles from the first w alleles in h_{s1} and the remaining (m_0-w) alleles from h_{s2} with $0 \leq w \leq m_0$. Similarly, there are another m_0+1 ways of forming h_{c1} from the first w alleles in h_{s2} and the remaining (m_0-w) alleles from h_{s1}. Let N_s be the set of feasible haplotype configurations $c = <h_{c1},h_{c2}>$ that can be inherited by child C_i from s of r with no more than one recombination. Thus, $|N_s| \leq 2(m_0+1)$. As h_{c2} is inherited from the haplotype configuration $q = <h_{q1}, h_{q2}>$ of F, let N_c be the set of feasible haplotype configurations of F which can produce the haplotype configuration c in C with no more than one recombination. Let $N'_{s,C_i} = \cup_{c\in N_s} N_c$, which indicates the set of feasible haplotype configurations in F which can go together with haplotype s in M to produce C_i. Obviously, $N'_{s,C_i} \leq 4(m_0+1)^2$.

As each haplotype configuration in F should be able to produce any of the children $C_1, \ldots, C_d$, the set of feasible haplotype configurations in F is $N'_s = \cap_i N'_{s,C_i}$. Equation (1) can be rewritten as:

$$num[r][s] = \min_{p \in N'_s} \{num[F][p] + \sum_i \min_{c_i \in N_s} (num[C_i][c_i] + numtrio(p,s,c_i))\} \quad (3)$$

As for Equation (2), if r is C_j and its haplotype configuration $s = <h_{s1}, h_{s2}>$, let $N_{s,F}$ and $N_{s,M}$ be the sets of feasible haplotype configurations in F and M, which can produce C_j with haplotype configuration s. As $|N_{s,F}| \leq 2(m_0+1)$ and $|N_{s,M}| \leq 2(m_0+1)$, let N_{p,C_i} (N_{q,C_i}) be the set of feasible haplotype configurations on another child C_i with haplotype configuration p in F (q in M) and $N''_{p,q} = N_{p,C_i} \cap N_{q,C_i}$ be the set of feasible haplotype configurations for each child C_i which can concurrently appear with the haplotype configuration s of child C_j. Note that $N''_{p,q} \leq 2(m_0+1)$ and Equation (2) can be rewritten as

$$\begin{aligned} num[r][s] = & \min_{p \in N_{s,F}, q \in N_{s,M}} \{numtrio(p,q,s) + num[F][p] + num[M][q] \\ & + \sum_{i \neq j} \min_{c_i \in N''_{p,q}} (num[C_i][c_i] + numtrio(p,q,c_i))\}, \text{where } r = C_j \end{aligned} \quad (4)$$

Theorem 1. *The 1-MRHI problem can be solved in $O(nm_0^4 2^{m_0})$ time and $O(n2^{m_0})$ space for a pedigree with n nodes and at most m_0 heterozygous loci in each node.*

Proof. Since we need $O(m_0)$ time to compute *numtrio* for each haplotype configuration combination in a trio and there are $8(m_0+1)^3$ haplotype configuration combinations in each trio, it takes $O(m_0^4 2^{m_0})$ time to process each trio. With at most n trios in the pedigree, time complexity is $O(nm_0^4 2^{m_0})$. Space complexity is $O(n2^{m_0})$ based on the size of *num*. ■

3.2 The *k*-MRHI Problem

Here we generalize the DP algorithm for the general *k*-MRHI problem by modifying the definition of neighboring haplotype configurations set from N_s to $N_s^{(k)}$: the set of haplotype configuration $c = <h_{c1}, h_{c2}>$ from s with no more than k recombinations. So we have $|N_s^{(k)}| = O(m_0^k)$. Similarly, we modify the definition of $N'^{(k)}_s = \cap_i N'_{s,C_i}{}^{(k)}$ with $N'_{s,C}$ to $N'_{s,C}{}^{(k)}$ in Equation (3) and the definition of $N_{s,F}$ and $N_{s,M}$ to $N_{s,F}{}^{(k)}$ and $N_{s,M}{}^{(k)}$, N_{p,C_i} to $N_{p,C_i}{}^{(k)}$, and $N''_{p,q}$ to $N''_{p,q}{}^{(k)}$ in Equation (4).

Theorem 2. *The k-MRHI problem can be solved in $O(nm_0^{3k+1} 2^{m_0})$ time for a pedigree with n nodes and at most m_0 heterozygous loci in each node.*

4 Root Selection for Better Performance

For 1-MRHI, the feasible haplotype configuration combinations in each trio may be much less than $O(m_0^3 2^{m_0})$ in practice because: (1) not all nodes have m_0

heterozygous loci; and (2) the number of feasible haplotype configurations a_v of a node v is also bounded by the number of feasible haplotype configurations a_{v_r} of v's neighbor v_r, *i.e.*, $a_v \le 2(\mu_v+1)a_{v_r}$, where μ_v is the number of heterozygous loci in v.

If v is M (or F, respectively), $\alpha_{C_i} = \min\{2^{\mu_{Ci}}, 2(\mu_{C_i}+1)\alpha_v\}$ ($i = 1, \ldots, k$) and $\alpha_F = \min_i\{2^{\mu_F}, 2(\mu_F+1)\alpha_{C_i}\}$. If v is C_i, then $\alpha_F = \min\{2^{\mu_F}, 2(\mu_F+1)\alpha_v\}$, $\alpha_M = \min\{2^{\mu_M}, 2(\mu_M+1)\alpha_v\}$ and $\alpha_{C_i} = \min\{2^{\mu_{Ci}}, 2(\mu_{C_i}+1)\ \alpha_F, 2(\mu_{C_i}+1)\ \alpha_M\}$ ($i = 1, \ldots, k$). Thus, the number of feasible haplotype configuration combinations, t_i in trio T_i can be computed consequently, assuming an arbitrary node (node R) as the root of the searching tree. The total number of feasible haplotype configuration combinations in all trios in the pedigree is $t_R = \Sigma_i t_i$, which can be computed in a tree traversal.

Theorem 3. *Let m_0 be the number of heterozygous loci and t_R be the total number of feasible haplotype configuration combinations for all trios in the pedigree with node R as root. Then the node which gives* $\min(t_R)$ *can be found in $O(n^2m_0)$ time and the 1-MRHI problem can be solved in $O(m_0 \min(t_R))$ time.*

The diameter of pedigree graphs in many practical instances is usually small. For example, the 452 families in the CEPH database [1][2][3] consist of only three generations. Suppose any node can be reached within l steps from R. We enumerate all 2^{μ_R} feasible haplotype configurations of the root, and no more than $2^{\mu_R} \times 2(m_0+1)$ feasible haplotype configurations for each of its neighboring nodes, and so on, and at most $2^{\mu_R} \times 2^l m_0^{\ l}$ at the most distant node.

Theorem 4. *The 1-MRHI problem can be solved in* $\min(O(nm_0^{\ 4}2^{m_0}), O(n2^{l+\mu_R}m_0^{\ l+1}))$ *time for a pedigree with n nodes and at most m_0 heterozygous loci in each node, where l is the maximum path length from the root to the leaves and μ_R is the number of heterozygous loci in root R.*

5 Experimental Results

We implemented the above DP algorithm in C++, and all experiments were conducted on a Pentium IV PC with 1.7GHz CPU and 256MB RAM.

5.1 Real Data

We examined real data set Epsiodic Ataxia (EA) by Litt et al.[10] which involves a family containing 29 people typed at 9 polymorphic markers on chromosome 12p. Both the locus-based algorithm [5] and the 1-MRHI algorithm run fast ($t < 1$ *sec.*) on this data set but the results are different. The locus-based algorithm gives a feasible solution with 5 recombinations in total but with a double recombination in one haplotype of member “100”. The 1-MRHI algorithm finds a more credible solution that has 6 recombinations in total, but with at most 1 recombination for each haplotype in the pedigree.

Another two real data sets are three generations families like those in the CEPH database [1][2][3] (*ftp://genome.wi.mit.edu/distribution/mpg/hapmap/ hap_struct/ popA/* (Gabriel et al.)): family 1331 on chromosome 7a, and family 1346 on chromosome 2a. After removing loci with missing alleles, family 1331 is a pedigree with 8 members on 32 loci, and family 1346 is a pedigree with 8 members on 55 loci. Both the locus-based and the 1-MRHI algorithm give the same answer for family 1331, but take 522.4s and 8.7s, respectively. As for family 1346, the locus-based algorithm fails because of not enough resources while the 1-MRHI algorithm finds a solution in 31 minutes.

5.2 Simulated Data

We compared our algorithm, with the locus-based algorithm [5] and PHASE [14], in terms of *running time t* and *accuracy ratio* ρ (in recovering the correct haplotype configurations for the whole pedigree). We used three different tree pedigree structures in the experiment: (1) a tree with 13 nodes (Figure 1), (2) a tree with 29 nodes (Figure 8 in [8]), and (3) a typical family with 21 nodes from the CEPH database [1][2][3].

Table 1. Comparison of performances of different haplotyping programs on simulation data

(n,r)	m = 15						m = 30	
	Locus-base		PHASE [13]		1-MRHI		1-MRHI	
	t (sec.)	ρ	*t (sec.)*	ρ*	*t (sec.)*	ρ	*t (sec.)*	ρ
(13, 0.0)	255.7	1.00	688.2	.87	1.68	1.00	202.8	1.00
(29, 0.0)	576.3	1.00	1772.8	.91	12.33	1.00	839.6	1.00
(21, 0.0)	234.4	1.00	592.4	.95	1.02	1.00	44.0	1.00
(13, 0.1)	287.7	.93	972.3	.85	1.73	.91	241.1	.92
(29, 0.1)	542.8	.90	2210.2	.90	10.45	.90	1042.8	.94
(21, 0.1)	243.2	.91	1504.2	.93	0.52	.94	33.7	.96
(13, 0.2)	294.2	.85	1221.4	.85	3.17	.89	1032.4	.86
(29, 0.2)	613.5	.81	3022.2	.89	11.70	.84	916.1	.84
(21, 0.2)	244.1	.90	2106.7	.93	1.22	.95	47.4	.92

[1] *Average performance is obtained from 100 independent executions of each program and for each parameter setting. n stands for the number of nodes, m for the number of marker loci, r for the recombination rate, t(sec.) for the average running time, and* ρ *for the accuracy ratio.*

[2] *The locus-based algorithm and PHASE cannot be applied where* $m \geq 30$ *due to space and time limitations, respectively.*

[*] *Accuracy ratio of PHASE is defined as the ratio of correctly resolved genotypes to all genotypes.*

For each pedigree, genotypes with 15 and 30 biallelic marker loci are considered. The two alleles at each locus of a founder are independently sampled with a fixed frequency of 0.5. The recombination rate is set to r = 0, 0.1, 0.2 between generations, and the number of recombinations is no more than one in each PO-pair.

As seen from Table 1, 1-MRHI runs quickest and can be applied to larger instances. All three algorithms can recover the correct haplotype configurations with high probability. The accuracy ratio decreases with the increase in the number of recombinations. Since we have limited the number of recombinations within each PO-pair to no more than one, the locus-based algorithm performs worse than the 1-MRHI algorithm as expected.

Acknowledgement

We thank T. Jiang and J. Li for sharing their DP code with us and the RGC Grant supporting this research.

References

[1] The CEPH genotype database. http://www.cephb.fr/.

[2] Dausset J. Cann H. Cohen D, Lathrop M, Lalouel J-M, White R. Centre d'Etude du Polymorphisme Humain (CEPH): collaborative genetic mapping of the human genome. Genomics. 6:575-577, 1990.

[3] Murray, J.C. et al. A comprehensive human linkage map with centimorgan density. Science, 265, 2049-2054 (1994).

[4] A.G. Clark. Inference of haplotypes from {PCR}-amplified samples of diploid populations. Mol. Biol. Evol, 7(2):111-122, 1990.

[5] K. Doi, J. Li, and T.Jiang. Minimum recombinant haplotype configuration on tree pedigree. In Proceedings of the Third International Workshop on Algorithms in Bioinformatics (WABI'03). Lecture Notes in Computer Science no. 2812, pages 339-353, 2003.

[6] Griffiths, W. Gelbart, R. Lewontin, and J. Miller. Modern Genetic Analysis: Integrating Genes and Genomes. W.H. Freeman and Company, N.Y., 2002.

[7] D. Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. J. Computational Biology, 8:305-323, 2001.

[8] J. Li and T. Jiang. Efficient inference of haplotypes from genotypes on a pedigree. J. Bioinfo Comp Biol, 1(1):41-69, 2003.

[9] J. Li and T. Jiang. Efficient rule-based haplotyping algorithms for pedigree data. In Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB'03), pages 197-206, 2003.

[10] M. Litt, P. Kramer, D. Browne, S. Gancher, E.R.P. Brunt, D. Root, et al. A gene for episodic ataxia/myokymia maps to chromosome 12p13. Am J Hum Genet 1994; 55: 702-709.

[11] J.R. O'Connell. Zero-recombinant haplotyping: applications to fine mapping using SNPs Genet Epidemiol, 19:S64-70, 2000.

[12] D. Qian and L. Beckmann. Minimum-recombinant haplotyping in pedigrees. Am J Hum Genet, 70(6): 1434-1445, 2002.

[13] E. Russo et al. Single nucleotide polymorphism: Big pharma hedges its bets. The Scientist, 13(15):1, 1999.

[14] M. Stephens, N.J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction for population data. Am. J. Hum. Genet, 68:978-989, 2001.

[15] P. Tapadar, S. Ghosh, and P.P. Majumder. Haplotyping in pedigrees via a genetic algorithm. Hum Hered, 50(1):43-56, 2000.