

Models and Algorithms for Haplotyping Problem

Xiang-Sun Zhang¹, Rui-Sheng Wang², Ling-Yun Wu¹ and Luonan Chen^{*,3}

¹*Institute of Applied Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China*

²*School of Information, Renmin University of China, Beijing 100872, China*

³*Department of Electrical Engineering and Electronics, Osaka Sangyo University, Osaka 574-8530, Japan*

Abstract: One of the main topics in genomics is to determine the relevance of DNA variations with some genetic disease. Single nucleotide polymorphism (SNP) is the most frequent and important form of genetic variation which involves a single DNA base. The values of a set of SNPs on a particular chromosome copy define a haplotype. Because of its importance in the studies of complex disease association, haplotyping is one of the central problems in bioinformatics. There are two classes of *in silico* haplotyping problems, i.e., single individual haplotyping and population haplotyping. In this review paper, we give an overview on the existing models and algorithms on this topic, report the recent progresses from the computational viewpoint and further discuss the future research trends.

1. INTRODUCTION

With complete genome sequences for humans now available, the investigation of genetic differences will be one of the main topics in genomics. In the genetic code transferring process or in the variation of health, slight mutations in nucleotides occasionally happen. For example, a G may have mutated into a T, and a nucleotide may be deleted or an additional one may be inserted. Such events give rise to the genetic variation of humans. Single nucleotide polymorphism (SNP) is a kind of genetic variation involving a single DNA base. According to Venter *et al.* [1], 2.1 million SNPs have been located on the whole human genome and less than 1% of all SNPs resulted in variations of proteins. SNP is the most frequent form [2] and is of importance in drug-design and medical applications. There are a large number of ongoing research works on not only determining SNP sites in human but also designing a detailed SNP map for human genome [3].

Human genomes are organized into pairs of chromosomes (a paternal copy and a maternal copy). The SNP sequence on each chromosome is called a *haplotype*, as illustrated in Fig. 1. The nucleotides in a SNP position are called *alleles*. In human, SNPs are almost always biallelic, i.e. there are two variants at the SNP sites which are denoted by 0 (wild type) and 1 (mutant type). Thus, a haplotype can be described as a string over $\{0,1\}$. A *genotype* is the conflation information of two haplotypes, i.e. a sequence of unordered allele pairs. For a genotype, when a pair of alleles at a SNP site is made of two identical values, this site is called *homozygous*, otherwise it is called *heterozygous*.

Haplotypes play a very important role in several areas of genetics, e.g. disease association studies and population history studies. However, it is substantially more difficult to

determine haplotypes than to determine genotypes or individual SNPs through biological experiments. On one hand, current sequencing techniques can only sequence at most several thousands base pairs as a whole. On the other hand, in terms of cost and labor, it is often only possible to detect the presence of SNP sites rather than to tell which copy of a pair of chromosomes the alleles belong to. Hence, computational methods that can reduce the cost of determining haplotypes, become attractive alternatives. They offer a way of inferring haplotypes from short DNA fragments with SNPs or from genotypes. There are two classes of *in silico* haplotyping problems: individual haplotyping and population haplotyping, in other words, haplotyping based on SNP fragments data or genotype samples. Individual haplotyping which is also called the haplotype *assembly* problem is to assemble the aligned SNP fragments from shotgun methodology, while population haplotyping is to infer a set of haplotypes of a population from their genotype data set, so it is called the haplotype *inference* problem.

There are already some review papers (e.g., Bonizzoni *et al.* [4], Gusfield [5], Halldörsson *et al.* [6], Stephens and Donnelly [7], Adkins [8]). Most of them have emphasized on either inference problem or assembly problem. Some of them concentrated on comparing algorithm details for a set of similar models. In this paper we try to give the readers a comprehensive overview on both assembly and inference problems from theoretical and computational viewpoints. First we display a general framework, on which most of the assembly models are designed. Biological assumptions underlying different models are clarified. Secondly we discuss the computational complexity of the models, the related algorithms and future trends, by reporting the recent progresses on this topic. Specifically, in Sections 2 and 3, we introduce the assembly problem and the inference problem separately, and conclusion is given in Section 4. Since this paper focuses on the review on current models and methods in a comprehensive manner, discussion of the detailed

*Address correspondence to this author at the Department of Electrical Engineering and Electronics, Osaka Sangyo University, Osaka 574-8530, Japan; E-mail: chen@elec.osaka-sandai.ac.jp

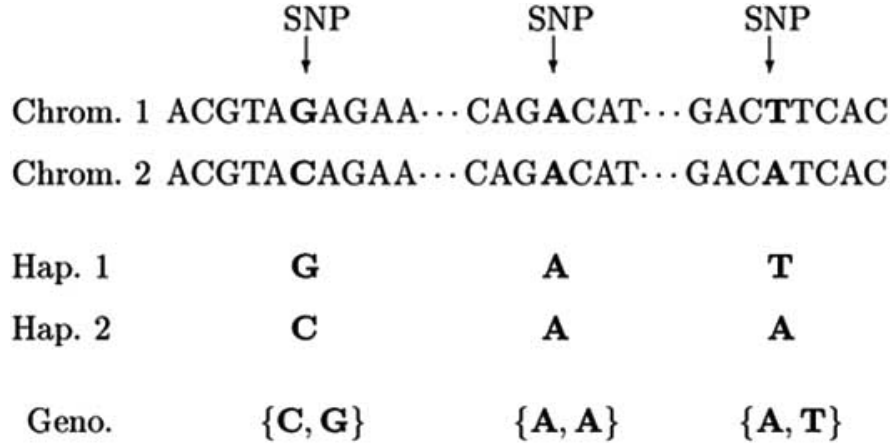


Fig. (1). The haplotypes and genotype of an individual. Chrom.: chromosome; Hap.: haplotype; Geno.: genotype.

algorithms and techniques is beyond scope of this paper and should be referred to each reference for detail description.

2. THE HAPLOTYPE ASSEMBLY PROBLEM

Single individual haplotyping or haplotype assembly, is based on data and methodology of shotgun sequence assembly [9,10]. The input data can be the aligned short DNA fragments with SNPs obtained by DNA shotgun sequencing or a resequencing effort for the purpose of large-scale haplotyping. When we focus on SNP positions, these short DNA fragments are actually aligned SNP fragments. Methods for haplotype assembly mainly solve such a basic problem: how aligned SNP fragments can be partitioned into two sets according to the SNP states, with each set determining a haplotype. Due to the current restrictive DNA sequencing techniques, only individual fragments or pairs of fragments which unavoidably contain sequencing errors (miscalled or skipped bases) are obtained. Moreover, DNA fragments possibly coming from other organisms may be wrongly mixed with the target one. In addition, these fragments may come from both copies of a pair of chromosomes and it is generally not easy to associate them to the right copy that they really belong to. All of these factors make haplotype assembly complicated. A formal and complete definition for the haplotype assembly problem is as follows (Lancia *et al.* [9]):

Given a set of inconsistent SNP fragments obtained by DNA sequencing, find and correct the errors in the data to retrieve a maximally consistent pair of haplotypes compatible with the corrected fragments.

Depending on different types of data errors, there are several different models in the literature for the problem. We will introduce these models below.

Suppose that there are m aligned SNP fragments from a pair of chromosomes and let the haplotypes' length be n . Define an $m \times n$ matrix $W = (w_{ij})$, whose row w_i represents for a SNP fragment and column for a SNP site s_j . The entry w_{ij} has value 0,1 or the hole symbol “-” (a missing base in the shotgun experiment or the entries that a fragment does not cover). Fig. 2 is an example of a 6x6 SNP matrix W . Let $x, y \in \{0,1,-\}$ and define

$$d(x,y) = \begin{cases} 1, & \text{if } x \neq -, y \neq - \text{ and } x \neq y, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Then, the distance of two fragments $w_i = (w_{i1}, \dots, w_{in})$ and $w_k = (w_{k1}, \dots, w_{kn})$ is defined as

$$d(w_i, w_k) = \sum_{j=1}^n d(w_{ij}, w_{kj}) \tag{2}$$

We say that two SNP fragments w_i and w_k are in *conflict* if $d(w_i, w_k) > 0$, otherwise we call them *compatible*. The distance between a fragment and a haplotype is similarly defined. A SNP fragment is said to be *gapless* if its 0s and 1s appear consecutively without “-” between them. A SNP matrix is gapless if all the SNP fragments in it are gapless.

Owing to the diploidy of human genome, if there is no error in the data, the rows of W can be divided into two disjoint sets (W_1, W_2) of compatible fragments. In this case, W is called *feasible* and (W_1, W_2) is a feasible partition. Two haplotypes can be inferred from the feasible partition sets respectively. In the general case, the matrix feasibility problem can be formulated into an OR (operation research)-nature model: the bipartite graph problem. Define a *conflict graph* $G = (V, E)$ for a given SNP matrix W , where $V = \{v_1, \dots, v_m$ (v_i represents w_i) is the vertex set and $E = \{(v_i, v_j) : d(w_i, w_j) > 0\}$ is the edge set. The conflict graph of a 6x6 SNP matrix is shown in Fig. 2. An important observation is that, W is feasible if and only if G is a bipartite graph. The following theoretical result about bipartite graph given by Skiena [11] is a basis for almost all established models in the assembly problem.

Theorem 1. (Skiena [11]) A graph is bipartite if and only if all its cycles are of even length, or there is no odd cycles in the graph.

By this theorem, intuitively, to convert a graph into a bipartite one we need to eliminate odd cycles, that is, to properly break some edges or remove some vertices on the odd cycles. Following this basic idea, various existing models were mainly deduced from the strategy of either cutting edges or removing vertices. We will review those models in the following section.

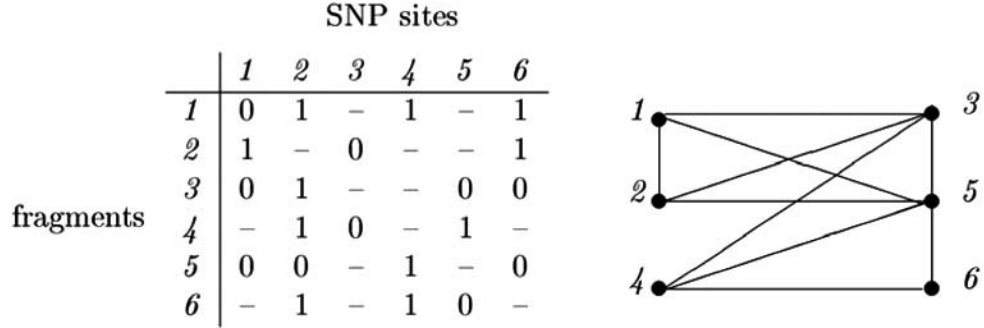


Fig. (2). A SNP matrix W and its fragment conflict graph.

2.1. Models Deduced from the Conflict Graph

2.1.1. Models Based on Error Fragment Removal

Removing a vertex on an odd cycle is equivalent to deleting a SNP fragment. Lancia *et al.* [9] suggested several ways to do that as follows:

- Minimum Fragment Removal (MFR)

This model assumes that “bad” fragments are due to contamination (fragments coming from another organism other than the target one) and thus focuses on removing error fragments.

Model Definition: Remove a minimum number of SNP fragments (rows of the SNP matrix) so that the resulting matrix is feasible.

Computational Complexity of the Model: MFR is NP-hard even for SNP matrices in which each fragment has at most one gap (Lancia *et al.* [9], Garey and Johnson [12]). Furthermore, Bafna *et al.* [13] have proved that MFR is APX-hard which indicates that there are no good approximation algorithms for it.

Algorithms: A dynamic programming algorithm with time complexity $O(2^{2k}m^2n+2^{3k}n^3)$ was given by Rizzi *et al.* [14], where k is the maximum number of holes in a fragment. Hence, MFR is fixed-parameter tractable, i.e., for small k the algorithm is practical. A fast heuristic algorithm for MFR was presented by Panconesi and Sozio [15].

- Longest Haplotype Reconstruction (LHR)

Model Definition: Remove a set of fragments so that the resulting matrix is feasible and the total length of the derived haplotypes is maximized.

Computational Complexity of the Model: LHR has polynomial-time algorithms when fragments are gapless, and is APX-hard in the 1-gap case (Cilibrasi *et al.* [16]).

Algorithms: Lancia *et al.* [9] gave a polynomial time algorithm for LHR when the SNP matrix is gapless and there is no fragment nested in another fragment. Cilibrasi *et al.* [16] introduced a dynamic programming algorithm for the gapless case with $O(n^2m+n^3)$ time order without other special assumption on fragments. However, there is no algorithm for the general case of this model in the literature to our knowledge.

2.1.2. Models Based on Conflicting Fragments Repair

Removing an edge on an odd cycle is equivalent to eliminating all conflicts between two fragments. There are

two ways to eliminate conflicts between two fragments: one is removing SNP sites that cause confliction, and the other is correcting (flipping) the SNP values of one fragment to the values of the conflicting fragment. The corresponding optimization problems are:

- Minimum SNP Removal (MSR)

This model assumes that all the fragments come from one organism but there are sequencing errors in the data. It aims to remove problematic SNP sites to make the resulting SNP fragments consistent.

Model Definition: Remove a minimum number of SNPs (columns of the SNP matrix) so that the resulting matrix is feasible.

Computational Complexity of the Model: MSR is NP-hard for SNP matrices with at most two gaps per fragment (Lancia *et al.* [9]). Recently, Bafna *et al.* [13] have further proved MSR to be APX-hard which indicates that there is no good approximation algorithm for it.

Algorithms: A dynamic programming algorithm with time complexity $O(mn2^{k+2})$ was given by Rizzi *et al.* [14], where k is the maximum number of holes in a fragment.

- Minimum Error Correction (MEC) or Minimum Letter Flips (MLF)

This model is also suited for the case that all the fragments come from one organism but there are sequencing errors. It differs from MSR on only correcting the SNP values but not deleting the problematic SNP site.

Model Definition: Correct a minimum number of entries so that the matrix is feasible, or flip a minimum number of letters to make the resulting matrix feasible.

Computational Complexity of the Model: The general MEC (MLF) problem was proved to be NP-hard by Lippert *et al.* [10]. Cilibrasi *et al.* [16] proved MEC to be NP-hard even if the SNP matrix is gapless by showing that MEC is a reduction from the MAX-CUT problem. They also proved MEC to be APX-hard in the 1-gap case. MEC was proved to be $O(\log n)$ -approximable by Panconesi and Sozio [15].

Algorithms: An exact algorithm based on branch-and-bound and a heuristic method based on genetic algorithm (GA) were proposed by Wang *et al.* [17].

- Weighted MLF (WMLF)

WMLF is motivated by a fact that when a sequencing machine sequences DNA fragments, it adds a confidence

information to each letter of these fragments. It attaches a weight, correlated with the confidence level of the letter flipped, to each flipping (Greenberg *et al.* [18]).

Model Definition: Given a set of aligned SNP fragments with confidence information, flip some letters so that the total weight of the flippings is minimized and the resulting SNP matrix is feasible.

Computational Complexity of the Model: WMLF is NP-hard even if the SNP matrix is gapless (Zhao *et al.* [19]).

Algorithms: A heuristic algorithm based on dynamic clustering was presented by Zhao *et al.* [19].

2.2. Algorithms Directly Based on the Conflict Graph

All the above excellent models are deduced from the conflict graph G , but they all do not make full use of the detailed information that G implies. In fact, G possesses useful information, such as distribution of odd cycles and the most crucial conflict pairs on the odd cycles, etc., which can be adopted to design an efficient algorithm. Moreover, existing graph theory and algorithms can be directly used to solve the haplotype assembly problem more professionally.

Along this idea, Reed *et al.* [20] and Hüffner [21] provided exact algorithms for the graph bipartization. Their algorithms seek for a minimal set of vertices by deletion to make the conflict graph bipartite. The algorithm given by Hüffner has time complexity $O(3^k \cdot |V| |E|)$, where k is the number of vertices to delete. They argued that as an exact algorithm the algorithm at the first glance is obviously exponential, but it is practical when the parameter k is small for given problems. This view is supported by the so called *fixed-parameter tractable* or *parameterized complexity* recently developed for dealing with the NP-complete or NP-hard problems (see Downey and Fellows [22]). Motivated by the works of Reed *et al.* and Hüffner, Guo *et al.* [23] recently considered deleting edges instead of vertices to make the graph bipartite. Their algorithm has time complexity $O(2^k \cdot |E|^2)$, where k is the number of edges to delete.

Hüffner [21] showed by real biological data that the haplotype assembly problem has reasonable size of k and the exact algorithm is practical. The importance of their work is to start designing exact algorithms to solve the assembly problem from the viewpoint of graph theory. It is noted that before the work of Reed *et al.* there has been few study in the field of graph theory to solve the optimal graph bipartization. It is the need in bioinformatics research that promotes new research works in graph theory.

2.3. The Hybrid Individual Haplotyping Model

By the word “hybrid”, we mean that in the haplotyping process of the assembly problem or the inference problem, both SNP fragments and genotype data are used. Generally speaking, the genotype information can be much more easily and economically obtained than the haplotype information. Hence, it is naturally to consider jointly using SNP fragments data and genotype data in the process of haplotyping. Along this direction, there are already some theoretical results. A hybrid method called Minimum Conflict Haplotyping (MCH) was proposed by Zhang *et al.* [24] for the individual haplotyping problem:

- Minimum Conflict Haplotyping (MCH)

Model Definition: Given an unphased genotype g and a set of SNP fragments of an individual, reconstruct a pair of haplotypes that is compatible with g and minimizes the number of conflicts with the given SNP fragments.

Computational Complexity of the Model: The MCH problem can be described as an integer linear programming and proved to be an NP-hard problem.

Algorithms: A dynamic programming procedure was introduced for one special case of MCH — there is no fragment completely covered by another fragment, and a feed-forward neural network was proposed to solve the model heuristically for general cases and large scale instances (Zhang *et al.* [24]).

2.4. Haplotype Assembly from a Statistical View

Li *et al.* [25] described a method for statistical reconstruction of haplotypes from a set of aligned SNP fragments. From a statistical view, haplotype assembly is:

Given a set of SNP fragments X (a SNP matrix), find a most probable pair of haplotypes among all possible pairs (h_1, h_2) , i.e. find a pair of haplotypes (h_1^, h_2^*) such that*

$$P(\{h_1^*, h_2^*\}) = \max_{h_1, h_2} P(\{h_1, h_2\} | X).$$

Li *et al.* [25] considered the matrix elements as random variables $X = (X_{ij})_{m \times n}$ and the underlying true bases as random variables $Y = (Y_{ij})_{m \times n}$. They introduced the haplotype composition variables $H = (H_{ij})_{2 \times n}$ and the fragment memberships $F = (F_i)$, $i = 1, 2, \dots, m$. The goal of their model is to evaluate the conditional distribution of the haplotype composition $P(H | X)$ or $P(H, F | X)$ over 2^{2n} possible alternatives, based on the true bases Y . Owing to the complexity of the problem, they used *Expectation-Maximization (EM)* algorithm (McLachlan and Krishnan [26]) to maximize $P(H | X)$ or $P(H, F | X)$.

2.5. Discussion

From above brief introduction we can see that all the models in Sections 2.1 and 2.2 stem from the transformation of a conflict graph into a spanning bipartite graph but consider different error types in SNP fragments. Which model we will apply in practice depends on the prior knowledge of the given data.

It is likely that most of the models for haplotype assembly are APX-hard, so there is no good constant factor approximation algorithm which is usually in combinatorial nature. Current research works display that many authors have turned their attention to heuristic methods which are out of combinatorial nature. Among these heuristic methods, genetic algorithm and artificial neural network (see general concept and algorithms in Goldberg [27] and Zhang [28]) are representative tools. Although these tools can not provide solutions with approximation ratio or accuracy guarantee, they have been theoretically proved to converge to local solutions of the problem — a concept in continuous optimization. They have been successfully applied to many areas including bioinformatics.

For algorithms without approximation guarantee, comparison between them needs extensive numerical experiments on well-established benchmark data sets. In this sense, to establish a complete series of benchmark data sets is a meaningful project in this research.

3. THE HAPLOTYPE INFERENCE PROBLEM

In this section, we introduce the population haplotyping problem. As defined above, a haplotype h is a vector (h_1, \dots, h_n) over $\{0,1\}$. A genotype g is a vector (g_1, \dots, g_n) over $\{0,1,2\}$. Let h_1, h_2 be a pair of haplotypes from the corresponding pair of chromosomes. Then the relationship between h_1, h_2 and g is:

$$\text{If } h_{1i} = h_{2i} \Leftrightarrow g_i = \begin{cases} 0, & h_{1i}, h_{2i} \text{ are wild} \\ 2, & h_{1i}, h_{2i} \text{ are mutant,} \end{cases}$$

$$\text{If } h_{1i} \neq h_{2i} \Leftrightarrow g_i = 1, \text{ the } i\text{th SNP site is heterozygous. (3)}$$

A pair of haplotypes h_1, h_2 is called a *haplotype configuration* or a *resolution* of a genotype g if they satisfy (3). We denote it by $h_1 \oplus h_2 = g$, $h_1 = g \ominus h_2$, $h_2 = g \ominus h_1$, where h_1, h_2 are said to *resolve* the genotype g . A genotype may have many haplotype configurations (2^{k-1} configurations if there are k heterozygous sites). A genotype is called *ambiguous* if it has at least two heterozygous positions, otherwise called *resolved*. A haplotype h is called *compatible* with a genotype g if for all $g_i \neq 2$, $h_i = g_i$. The genotype data in a population of size m can be formulated as an $m \times n$ matrix $G = \{g_{ij}\}$ on $\{0,1,2\}$ with each row g_i corresponding to a genotype and each column j corresponding to a SNP site on the chromosome. A *realization* of a genotype matrix is a haplotype matrix H on $\{0,1\}$ with each row corresponding to a haplotype, and for each genotype g_i there are two rows (a pair of haplotypes) h_1, h_2 of H such that h_1, h_2 form a resolution of g_i .

For a pair of chromosomes of a child, if no *recombination* occurs, one copy is identically inherited from the paternal genome and the other from the maternal genome, otherwise, portions of the paternal or maternal chromosomes are exchanged during inheritance, as shown in Fig. 3. Biological experiments [29] show that human chromosomes have block structure called *linkage disequilibrium* block, where within each block no or few recombinations could occur and haplotypes have very low diversity. These facts make the haplotype mapping and disease association study possible. Haplotype inference is to resolve the heterozygous sites in a set of genotype data, i.e. to determine which copy of a pair of chromosomes each allele belongs to. Specifically, the haplotype inference problem is:

Given an $m \times n$ genotype matrix G , find a haplotype matrix H such that for each genotype there exists at least one pair of haplotypes in H which is a resolution of this genotype.

As well known, without any biological insight or genetic model, we cannot infer haplotypes from genotype data, because there may be an exponential number of possible haplotype configurations. If we arbitrarily select a pair of haplotypes among them for a genotype, the haplotype inference problem is trivial and we do not know which one is

“true”. Blocks of limited haplotype diversity make haplotype inference somewhat easier than the case that many recombination events occur. Therefore, an inferring method mainly considers the analysis of a specific block in the population. Actually, the haplotype inference problem has several versions based on different genetic models.

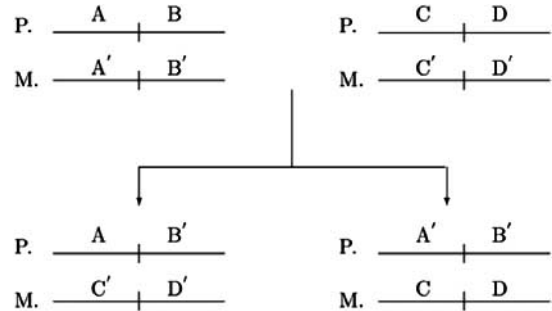


Fig. (3). Recombination. A recombination event occurs when the chromosomes of the left child are inherited from parents, and no recombination occurs when the chromosomes of the right child are inherited from parents, where P. denotes the paternal copy of a pair of chromosomes, and M. denotes the maternal copy of a pair of chromosomes.

3.1. Haplotype Inference by Parsimony

Clark [30] first pointed out some basic computational issues related to haplotype inference under a general *inference rule*. Suppose that G is a set of genotypes with n SNP sites. First, all the resolved genotypes in G are identified and form a haplotype set H . For a genotype $g \in G$, select a haplotype h compatible with g in H . If there is no $g \ominus h$ in H , add $g \ominus h$ into H . Remove g from G . The process continues until either all genotypes are resolved or no haplotype in H is compatible with the left genotypes. They applied a greedy approach when the known haplotypes are tested against the unresolved genotypes. Clark’s experiments on real data and simulated data suggest that a valid solution is usually the one resolves a maximum number of genotypes. Gusfield [31] further studied a parsimony version of this problem.

- Maximum Resolution (MR)

This model is based on Clark’s inference rule and a parsimony principle.

Model Definition: Given a set of genotypes G (including some resolved ones), find a maximum number of genotypes that can be resolved by successive application of Clark’s inference rule.

Computational Complexity of the Model: The MR model can be exactly formulated as an integer linear programming and was proved to be NP-hard (Gusfield [31]).

Algorithms: Gusfield [31] and [32] employed a graph-theoretical method to express and analyze the inference problem. Linear programming relaxation is adopted to solve the practical problem.

- Single Genotype Resolution (SGR)

This model was formulated by Bonizzoni *et al.* [4]. It is also based on Clark's inference rule and related to the MR problem.

Model Definition: Given a nonempty set of haplotypes H and a distinguished genotype g in a set of genotypes G , find a sequence of applications of the Clark's inference rule that resolves a subset $G_0 \subset G$ including g or conclude that such a sequence does not exist.

Computational Complexity of the Model: The computational complexity of the SGR problem was listed as an open problem by Bonizzoni *et al.* [4]. Lin *et al.* [33] solved it by reduction from 3-SAT problem and thus proved that the SGR problem is NP-complete.

Algorithms: So far there is no algorithm designed for this problem.

- Haplotype Inference by Pure Parsimony (HIPP)

The pure parsimony criterion for haplotype inference was proposed by Gusfield [34]. Its reasonability and biological meanings were illustrated in Gusfield [34], Wang and Xu [35]. This criterion is based on the fact that in natural populations, the number of the observed distinct haplotypes is vastly smaller than the number of combinatorially possible haplotypes.

Model Definition: Given a set of genotypes G , find a cardinality-smallest set of haplotypes H such that for each $g \in G$, there is a haplotype configuration consisting of two sequences in H to resolve g .

Computational Complexity of the Model: The HIPP problem is APX-hard (Lancia *et al.* [36]).

Algorithms: A branch-and-bound method was suggested by Wang and Xu [35]. An integer programming model of exponential size was presented in Gusfield [34], and later an integer programming model of linear size was given by Brown and Harrower [37]. A first approximation algorithm with performance guarantee 2^{k-1} was designed by Lancia *et al.* [36] for the case in which each genotype has at most k heterozygous positions. Huang *et al.* [38] proposed an $O(\log n)$ -approximation algorithm, where n is the number of genotypes. A heuristic algorithm — Parsimonious Tree-Grow method (PTG) in $O(m^2n)$ time (m is the number of SNP sites and n is the number of genotypes) was developed by Li *et al.* [39], which can not only solve haplotyping problem in an accurate and efficient manner but also numerically handle large-scale problems, and a software is also provided for PTG in <http://zhanggroup.aporc.org/bioinfo/ptg/>. Recently, Lancia and Rizzi [40] presented a polynomial time algorithm for the HIPP problem when each genotype has at most two heterozygous positions.

3.2. Perfect Phylogeny Haplotyping

Perfect phylogeny haplotyping (PPH) is based on the *coalescent* theory in genetics, i.e. the evolutionary history of the haplotypes in the population can be described by a rooted tree and each haplotype is a leaf of the tree. Another assumption of PPH is *infinite site*. That is to say, at most one mutation occurs in a given site in the tree and recurrent mutations are forbidden. Hence, the infinite site model is suitable for describing the evolutionary history without recombination.

Let H be a set of haplotypes (an $m \times n$ matrix on $\{0,1\}$). A *haplotype perfect phylogeny* for H is a rooted tree with the following properties:

- (1) Each leaf in the tree denotes a distinct haplotype in H ;
 - (2) Each edge represents a SNP site with a mutation from 0 to 1, and each site is labeled by at most one edge;
 - (3) For each haplotype labeled by a leaf of the tree, the unique path from the root to itself specifies all SNP sites with value 1 in this haplotype.
- Perfect Phylogeny Haplotype (PPH)

Model Definition: Given a genotype matrix G , find a haplotype set H such that for each $g \in G$, there is a pair of haplotypes in H acting as a resolution of g and H has a haplotype perfect phylogeny; otherwise conclude that such a matrix does not exist.

Computational Complexity of the Model: The PPH problem is polynomially solvable.

Algorithms: Gusfield [41] solved the PPH problem by transforming it into a graph realization problem. Though the algorithm is polynomial time of $O(mn^2)$, its realization procedure is very complicated. Several direct polynomial algorithms were independently proposed by Bafna *et al.* [42] and Eskin *et al.* [43]. Bafna *et al.* [44] claimed that the above algorithms can not be adapted to an algorithm of $O(mn)$ and any linear time solution to the PPH problem likely requires a different approach.

- Minimum Perfect Phylogeny Haplotype (MPPH)

Model Definition: Given a genotype matrix G , find a haplotype set H with smallest cardinality such that for each $g \in G$, there is a pair of haplotypes in H acting as a resolution of g and H has a haplotype perfect phylogeny; otherwise decide that such a matrix does not exist.

Computational Complexity of the Model: Bafna *et al.* [44] showed that the problem is NP-hard by a reduction from Vertex Cover.

Algorithms: So far there is no algorithm designed for the MPPH problem.

When there is inconsistency (read-errors, missing bases or an imperfect fit to the perfect phylogeny model) in the data, PPH becomes a computationally hard problem. Kimmel and Shamir [45] proved that the perfect phylogeny haplotype problem is NP-complete when some of the data entries are missing. Fortunately, Gramm *et al.* [46] found that haplotyping *via* perfect phylogeny with missing data becomes computationally tractable when imposing additional biologically motivated constraints. Halperin and Eskin [47] successfully used imperfect phylogeny model to reconstruct long haplotypes. The computational complexity of PPH with recombination is still an open problem.

3.3. Statistical Models for Haplotype Inference

In the statistical models for haplotype inference, there is an underlying unknown distribution of the haplotype frequencies in the population. These models often assume the Hardy-Weinberg equilibrium (HWE). That is, the

probability of a genotype is related to the probabilities of its haplotype configurations. In detail, for a genotype $g \in G$,

$$Pr(g) = \sum_{h \oplus \bar{h} = g} Pr(h)Pr(\bar{h})$$

where Pr represents the probability. The *maximum likelihood method* tries to estimate the haplotype frequencies that maximize the likelihood function of the observed genotype set G . This problem is also known as *haplotype frequency estimation*. An *Expectation-Maximization* (EM) algorithm was proposed by Excoffier *et al.* [48]. Note that since there is an exponential number of possible haplotypes, the EM algorithm can not handle satisfactorily the problem with long haplotypes. Recently, a *Partition-Ligation-Estimation-Maximization* algorithm developed by Niu *et al.* [49] uses a *divide and conquer* approach to address this issue.

Stephens *et al.* [50] proposed a modification of maximum likelihood method by introducing an approximate population genetic model to generate a priori distribution of the haplotype frequencies. Then they try to find the posterior distribution of the haplotype frequencies for a given genotype set G . This problem is called *Bayesian haplotype inference*. Stephens *et al.* solved the problem by a Markov Chain Monte Carlo (MCMC) approach. A partition-ligation variant of MCMC approach was proposed by Niu *et al.* [51] to deal with large-scale problems.

Instead of estimating frequencies of full haplotypes as above models, the Markov chain model proposed by Eronen *et al.* [52] estimates and uses frequencies of short haplotype fragments. The haplotype frequencies are calculated from the fragment frequencies by modeling the haplotype as a Markov chain of short fragments. The Markov chain model can better adapt to the recombination since it does not assume haplotype blocks. In order to find the haplotype configuration with maximum likelihood from solution space with exponential number of haplotypes, Eronen *et al.* use a heuristic partition-ligation method like that in Niu *et al.* [49]. Recently, Zhang *et al.* [53] proved that this problem can be exactly solved by a dynamic programming algorithm with polynomial time complexity.

Besides the Markov chain model, models explicitly considering the recombination include recent works of Greenspan *et al.* [54] and Kimmel *et al.* [55]. The model of Greenspan *et al.* is based on Bayesian network, while the model of Kimmel *et al.* follows the maximum likelihood method.

When statistical models are adopted, probability and stochastic process theory play main roles in establishing models and designing algorithms. But one can also find in the literature, an efficient algorithm may require both deterministic and stochastic techniques, depending on the problem setting. Since there are an exponential number of feasible solutions in the haplotyping problem, to design an efficient computational method with a high accuracy is still an important task.

3.4. Haplotype Inference in Pedigree

All above methods for haplotype inference focus on haplotyping in unrelated population. In contrast, another group of methods is based on pedigree data. Haplotype

inference based on pedigree data has two fundamental assumptions: (1) the given genotype data has a pedigree structure called pedigree graph. That is to say, the individuals in the population are genetically related; (2) the inheritance satisfies the Mendelian law, i.e. out of two alleles in every SNP site of the genotype of a child, one comes from his paternal genome and the other from his maternal genome, and there is no mutation to occur during the inheritance. One then can get a better estimation of haplotypes because the haplotypes of a child is constrained by its inheritance from his parents. However, collection of such pedigree data (related individuals) costs much more than that of unrelated population. This version of haplotype inference is also considered in many literatures, such as Li *et al.* [56,57] and Doi *et al.* [58]. The models are based on the fact that few recombinations occur when the haplotypes of a child inherit from parents, and hence the objective is often to minimize the total number of recombinations.

- Minimum Recombination Haplotype Configuration (MRHC)

Model Definition: Given a valid genotype pedigree graph G , find a realization H of G involving a minimum number of recombination events.

Computational Complexity of the Model: Li *et al.* [56] proved MRHC on pedigree graph to be NP-hard by reduction from 3-Dimensional Matching. Doi *et al.* [58] proved MRHC on pedigree tree is also NP-hard.

Algorithms: Algorithms based on rules were presented by Tapadar *et al.* [59] and Qian and Bechkmann [60]. An iterative heuristic algorithm was proposed by Li *et al.* [56] for MRHC on pedigree graph. Doi *et al.* [58] gave two dynamic programming algorithms for MRHC on pedigree tree.

- Zero Recombination Haplotype Configuration (ZRHC)

Model Definition: Given a valid genotype pedigree graph G , find a realization H of G involving no recombination events or decide that such realization does not exist.

Computational Complexity of the Model: ZRHC is polynomial time solvable (Li *et al.* [56]).

Algorithms: Li *et al.* [56] presented an algorithm based on Gaussian elimination for ZRHC.

- k -Minimum Recombination Haplotype Configuration (k -MRHC)

Model Definition: Given a valid genotype pedigree graph G , find a realization H of G such that the total number of recombinations is minimal and the number of recombinations on each parent-offspring pair is at most k .

Computational Complexity of the Model: Chin *et al.* [61] proved k -MRHC on pedigree graph to be NP-hard even for $k = 1$. The computational complexity of k -MRHC on pedigree tree is still open.

Algorithms: A dynamic programming algorithm in $O(nm_0^{3k+1} 2^{m_0})$ time on pedigrees with n nodes and at most m_0 heterozygous loci in each node was proposed by Chin *et al.* [61].

Several variants of MRHC were also formulated by Bonizzoni *et al.* [4]. The complexity and algorithms of these variants are mostly open.

3.5. Discussion

There is no such a unified framework for the haplotype inference problem as in the haplotype assembly, where the conflict graph and its bipartization take a key role. There are parallel two main methodologies in inference model formulation: one is deterministic and the other is statistic. Various inference models have been suggested, by featuring different computational complexity ranging from P to NP.

These models of haplotype inference have different biological assumptions. Therefore theoretical comparison of methods with different assumptions is difficult, if not impossible. Some researchers have done comparison of haplotype inference methods based on similar models. For example, Stephens and Donnelly [7] compared three Bayesian methods, emphasizing the differences between the models and the computational strategies. Numerical comparison of models and algorithms need large and objective data sets since every model has its suited data. Extensive and objective accuracy comparison of them is a heavy technical but important research work. There are already some attempts, e.g. Adkins [8] compared the accuracy of some methods using a large set of 308 empirically determined haplotypes based on 15 SNPs. Evaluating all the existing inference models on one framework and unified data sets is still a future study problem.

In practice, the optimal solution of the haplotype inference problem does not necessarily correspond to "true" haplotypes with 100% accuracy. Like the hybrid individual haplotyping model, incorporating some other information (e.g. SNP fragments) into existing models may improve the accuracy of haplotype inference. Developing such a hybrid model and designing algorithms for it will be a future research direction.

4. CONCLUSION

In this paper, we introduce a very important computational problem in bioinformatics, i.e. the haplotyping problem, by reviewing recent progress in this area. There are two basic haplotyping problems: individual haplotyping and population haplotyping. The individual haplotyping is a process of reconstructing a pair of haplotypes of an individual from a set of SNP fragments, whereas the population haplotyping is to generate a set of haplotypes for resolving a set of genotypes from a population. In this review, different models for both assembly and inference problems are briefly introduced.

It is noted that the conflict graph takes a significant role in the assembly models and algorithms. In particular, we show that almost all of the existing models are related to bipartite graph, which not only gives graphical explanation of each model but also can be used to derive an efficient algorithm for the haplotyping problem. In other words, deterministic optimization theory and methods are essential tools in this area. The progress of the assembly problem history can be divided into two stages. In the first stage, most

heuristic (approximate) algorithms are based on the concept of conflict graph without using the concrete information included in the conflict graph. In the second stage one designs exact algorithms by turning a conflict graph to a bipartite graph. The designed exact algorithms are exponential in nature but may be feasible for the practical problems.

Since most of the models for haplotype assembly are APX-hard, we have to adopt heuristic algorithms without accuracy guarantee to solve them. Then, to establish a set of benchmark data sets for testing turns to be an important project. Benchmark data sets should come from both real biological experiments and theoretical simulation data.

The studies of haplotype inference problem started with well-known Clark's rule. Various combinatorial methods to infer haplotypes from genotype data were developed from or closely related to Clark's rule. By biological assumption, the parsimony model and coalescent model were introduced to make the problem more realistic and efficient to solve. To further explore the inherent content in a given genotype set, many researchers turned their attention to the statistical nature of the inference problem. It is because that a population with biological definition must provide statistical information. With this observation, various probability models and stochastic process are used to build inference models. Recent research [6] indicated that the statistical models for haplotype inference are also linked to the combinatorial models. Moreover, deterministic algorithms, including combinatorial algorithms and heuristic algorithms, are used to solve statistical models. All those works display the multidisciplinary nature of the bioinformatics research.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under grant No. 10471141. The work of Prof. Zhang is partly supported by the Informatics Research Center for Development of Knowledge Society Infrastructure, Graduate School of Informatics, Kyoto University, Japan.

REFERENCES

- [1] Venter JC, Adams MD, Myers EW, *et al.* The sequence of the human genome. *Science* **2001**; 291(5507): 1304–1351.
- [2] Chakravarti A. It's raining, hallelujah? *Nat Genet* **1998**; 19: 216–217.
- [3] Altshuler D, Pollarn VJ, Cowles CR, *et al.* An SNP map of the human genome generated by reduced representation shotgun sequencing. *Nature* **2000**; 407: 513–519.
- [4] Bonizzoni P, Vedova GD, Dondi R, Li J. The haplotyping problem: An overview of computational models and solutions. *J Compu Sci Tech* **2003**; 18(6): 675–688.
- [5] Gusfield D. An overview of combinatorial methods for haplotype inference. In Proceedings of the 1st RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotype Inference. Springer-Verlag GmbH 2004; 9–25.
- [6] Halldórsson BV, Bafna V, Edwards N, Lippert R, Yooshep S, Istrail S. A survey of computational methods for determining haplotypes. In Proceedings of the 1st RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotype Inference. Springer-Verlag GmbH 2004; 26–47.
- [7] Stephens M, Donnelly P. A comparison of bayesian methods for haplotype reconstruction from population genotype data. *The Am J Hum Genet* **2003**; 73: 1162–1169.

- [8] Adkins RM. Comparison of the accuracy of methods of computational haplotype inference using a large empirical dataset. *BMC Genet* **2004**; doi:10.1186/1471-2156-5-22.
- [9] Lancia G, Bafna V, Istrail S, Lippert R, Schwartz R. SNPs problems, complexity and algorithms. In Proceedings of Annual European Symposium on Algorithms (ESA), Volume 2161 of Lecture Notes in Computer Science. Springer 2001; 182–193.
- [10] Lippert R, Schwartz R, Lancia G, Istrail S. Algorithmic strategies for the SNPs haplotype assembly problem. *Briefings in Bioinformatics* **2002**; 3(1): 23–31.
- [11] Skiena S. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematics, chapter Coloring Bipartite Graphs, 213. Addison-Wesley, Reading, MA 1990.
- [12] Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company 1979.
- [13] Bafna V, Istrail S, Lancia G, Rizzi R. Polynomial and APX-hard cases of the individual haplotyping problem. *Theoret Compu Sci* **2005**; 335: 109–125.
- [14] Rizzi R, Bafna V, Istrail S, Lancia G. Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem. In Guigo R, Gusfield D, editors, Proceedings of 2nd Annual Workshop on Algorithms in Bioinformatics (WABI), Volume 2452 of Lecture Notes in Computer Science. Springer 2002; 29–43.
- [15] Panconesi A, Sozio M. Fast hare: A fast heuristic for single individual SNP haplotype reconstruction. In Proceedings of the 4th Annual International Workshop on Algorithms in Bioinformatics (WABI). Springer-verlag 2004; 266–277.
- [16] Cilibrasi R, Iersel L, Kelk S, Tromp J. On the complexity of the single individual haplotyping problem. Proceedings of WABI2005 2005.
- [17] Wang RS, Wu LY, Li ZP, Zhang XS. Haplotype reconstruction from SNP fragments by minimum error correction. *Bioinformatics* **2005**; 21(10): 2456–2462.
- [18] Greenberg HJ, Hart WE, Lancia G. Opportunities for combinatorial optimization in computational biology. *INFORMS J Comput* **2004**; 14(1): 211–231.
- [19] Zhao YY, Wu LY, Zhang JH, Wang RS, Zhang XS. Haplotype assembly from aligned weighted SNP fragments. *Comput Biol Chem* **2005**; 29(4): 281–287.
- [20] Reed B, Smith K, Vetta A. Finding odd cycle transversals. *Operat Res Letters* **2004**; 32(4): 299–301.
- [21] Hüffner F. Algorithm engineering for optimal graph bipartization. In Proceedings of the 4th International Workshop of Efficient and Experimental Algorithms (WEA). Springer-Verlag 2005; 240–252.
- [22] Downey RG, Fellows MR. Parameterized Complexity. Springer 1999.
- [23] Guo J, Gramm J, Hüffner F, Niedermeier R, Wernicke S. Improved fixed parameter algorithms for two feedback set problems. In Proceedings of the 9th Workshop on Algorithms and Data Structures (WADS). Springer-Verlag **2005**; 158–168.
- [24] Zhang XS, Wang RS, Wu LY, Zhao YY, Chen L. Minimum conflict haplotyping from genotype data and SNP fragments. Report of Research IAM-R001, Institute of Applied Mathematics, Academy of Mathematics and Systems Science, CAS 2005.
- [25] Li LM, Kim JH, Waterman MS. Haplotype reconstruction from SNP alignment. *J Comput Biol* **2004**; 11: 507–518.
- [26] McLachlan GJ, Krishnan T. The EM Algorithm and Extensions. Wiley, New York 1997.
- [27] Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading: Addison-Wesley 1989.
- [28] Zhang XS. *Neural Networks in Optimization*, Kluwer Academic Publishers 2000.
- [29] Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES. High-resolution haplotype structure in the human genome. *Nature Genetics* **2001**; 29: 229–232.
- [30] Clark AG. Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol Biol Evolut* **1990**; 7(2): 111–122.
- [31] Gusfield D. Inference of haplotypes from samples of diploid populations: Complexity and algorithms. *J Comput Biol* **2001**; 8(3): 305–323.
- [32] Gusfield D. A practical algorithm for optimal inference of haplotypes from diploid populations. In Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB). AAAI Press 2000; 183–189.
- [33] Lin H, Zhang Z, Zhang Q, Bu D, Li M. A note on the single genotype resolution problem. *J Comput Sci Tech* **2004**; 19(2): 254–257.
- [34] Gusfield D. Haplotyping by pure parsimony. In Proceedings of the 14th Symposium on Combinatorial Pattern Matching (CPM). Springer-Verlag GmbH 2003; 144–155.
- [35] Wang LS, Xu Y. Haplotype inference by maximum parsimony. *Bioinformatics* **2003**; 19(14): 1773–1780.
- [36] Lancia G, Pinotti C, Rizzi R. Haplotyping population by pure parsimony: Complexity of exact and approximation algorithms. *INFORMS J Comput* **2004**; 16(4): 438–359.
- [37] Brown DG, Harrower IM. A new integer programming formulation for the pure parsimony problem in haplotype analysis. In Proceedings of the 4th International Workshop on Algorithms in Bioinformatics (WABI) 2004; 254–265.
- [38] Huang YT, Chao KM, Chen T. An approximation algorithm for haplotype inference by maximum parsimony. In Proceedings of the 2005 ACM Symposium on Applied Computing 2005; 146–150.
- [39] Li ZP, Zhou WF, Zhang XS, Chen L. A parsimonious tree-grow method for haplotype inference. *Bioinformatics* **2005**; 21(17): 3475–3481.
- [40] Lancia G, Rizzi R. A polynomial case of the parsimony haplotyping problem. *Operat Res Letters* **2005**; in press.
- [41] Gusfield D. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In Proceedings of 6th Annual International Conference on Research in Computational Molecular Biology (RECOMB). ACM Press **2002**; 166–175.
- [42] Bafna V, Gusfield D, Lancia G, Yooseph S. Haplotyping as a perfect phylogeny: A direct approach. *J Comput Biol* **2003**; 10(3): 323–340.
- [43] Eskin E, Halperin E, Karp R. Efficient reconstruction of haplotype structure via perfect phylogeny. *J Bioinform Comput Biol* **2003**; 1(1): 1–20.
- [44] Bafna V, Gusfield D, Hannehalli S, Yooseph S. A note on efficient computation of haplotypes via perfect phylogeny. *J Comput Biol* **2004**; 11(5): 858–866.
- [45] Kimmel G, Shamir R. The incomplete perfect phylogeny haplotype problem. *J Bioinform Comput Biol* **2005**; 3(2): 359–384.
- [46] Gramm J, Nierhoff T, Tantau T. Perfect path phylogeny haplotyping with missing data is fixed-parameter tractable. In Lecture Notes in Computer Science, 3162. Springer-Verlag 2004; 174–186.
- [47] Halperin E, Eskin E. Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics* **2004**; 20(12): 1842–1849.
- [48] Excoffier L, Slatkin M. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol Biol Evolut* **1995**; 12(5): 921–927.
- [49] Niu T, Qin ZS, Liu JS. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *The Am J Hum Genet* **2002**; 71: 1242–1247.
- [50] Stephens M, Smith NJ, Donnelly P. A new statistical method for haplotype reconstruction from population data. *The Am J Hum Genet* **2001**; 68: 978–989.
- [51] Niu T, Qin ZS, Xu X, Liu JS. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *The Am J Hum Genet* **2002**; 70: 157–169.
- [52] Eronen L, Geerts F, Toivonen H. A markov chain approach to reconstruction of long haplotypes. In Proceedings of 9th Pacific Symposium on Biocomputing (PSB'04). World Scientific 2004; 104–115.
- [53] Zhang JH, Wu LY, Zhang XS. A dynamic programming method for haplotype inference based on Markov chain model. Working paper, Institute of Applied Mathematics, Academy of Mathematics and Systems Science, CAS 2005.
- [54] Greenspan G, Geiger D. Model-based inference of haplotype block variation. In Proceedings of the 7th Annual International Conference on Computational Molecular Biology (RECOMB) 2003; 131–137.
- [55] Kimmel G, Shamir R. Maximum likelihood resolution of multi-block genotypes. In Proceedings of the 8th Annual International Conference on Computational Molecular Biology (RECOMB) 2004.
- [56] Li J, Jiang T. Efficient inference of haplotype from genotype on a pedigree. *J Bioinform Comput Biol* **2003**; 1(1): 41–69.

- [57] Li J, Jiang T. An exact solution for finding minimum recombinant haplotype configurations on pedigrees with missing data by integer linear programming. In Proceedings of the 8th Annual International Conference on Research in Computational Molecular Biology (RECOMB). ACM press 2004; 20–29.
- [58] Doi K, Li J, Jiang T. Minimum recombinant haplotype configuration on tree pedigrees. In Proceedings of the 3th Annual International Workshop on Algorithms in Bioinformatics (WABI). Springer-verlag GmbH 2003; 339–353.
- [59] Tapadar P, Ghosh S, Majumder PP. Haplotyping in pedigrees via a genetic algorithm. *Hum Hered* **2000**; 50(1): 43–56.
- [60] Qian D, Beckmann L. Minimum-recombinant haplotyping in pedigrees. *The Am J Hum Genet* **2002**; 70(6): 1434–1445.
- [61] Chin FY, Zhang QF, Shen H. *k*-recombination haplotype inference in pedigrees. In Proceedings of the International Conference on Computational Science (ICCS), LNCS 3515. Springer-Verlag, Berlin Heidelberg 2005; 985–993.

Received: June 29, 2005

Revised: July 23, 2005

Accepted: July 26, 2005