An Approximation Algorithm for Haplotype Inference by Maximum Parsimony

(Class note)

Yao-Ting Huang,[1] Kun-Mao Chao,[1,2,†] and Ting Chen[3,†]

[1]Department of Computer Science and Information Engineering

[2]Graduate Institute of Networking and Multimedia

National Taiwan University, Taipei, Taiwan

{d92023, kmchao}@csie.ntu.edu.tw

[3]Department of Biological Sciences

University of Southern California, Los Angeles, CA 90089, USA

tingchen@usc.edu

†: Corresponding Authors:

Kun-Mao Chao

Department of Computer Science

and Information Engineering

National Taiwan University

♯1 Roosevelt Rd. Sec. 4, Taipei, Taiwan

Email: kmchao@csie.ntu.edu.tw

Phone: 886-2-23625336

Fax: 886-2-23628167

Ting Chen

Department of Biological Sciences

University of Southern California

1042 West 36th Place, DRB 290

Los Angeles, CA 90089-1113, USA

Email: tingchen@usc.edu

Phone: 213-740-2415

Fax: 213-740-2437

Abstract

This paper studies haplotype inference by maximum parsimony using population data. We define the optimal haplotype inference (OHI) problem as given a set of genotypes and a set of related haplotypes, find a minimum subset of haplotypes that can resolve all the genotypes. We prove that OHI is NP-hard and can be formulated as an integer quadratic programming (IQP) problem. To solve the IQP problem, we propose an iterative semi-definite programming based approximation algorithm, (called SDPHapInfer). We show that this algorithm finds a solution within a factor of $O(\log n)$ of the optimal solution, where n is the number of genotypes. This algorithm has been implemented and tested on a variety of simulated and biological data. In comparison with three other methods: (1) HAPAR, which was implemented based on the branching and bound algorithm, (2) HAPLOTYPER, which was implemented based on the Expectation-Maximization algorithm, and (3) PHASE, which combined the Gibbs sampling algorithm with an approximate coalescent prior, the experimental results indicate that SDPHapInfer and HAPLOTYPER have similar error rates. In addition, the results generated by PHASE have lower error rates on some data but higher error rates on others. The error rates of HAPAR are higher than the others on biological data. In terms of efficiency, SDPHapInfer, HAPLOTYPER, and PHASE output a solution in a stable and consistent way, and they run much faster than HAPAR when the number of genotypes becomes large.

Keywords: algorithm, haplotype inference, integer quadratic programming, maximum parsimony, semi-definite programming

1 Introduction

Correlating variations in DNA sequence with phenotypic differences has been one of the grand challenges in biology. Efforts have been made to obtain all common variants in the human population, including single nucleotide polymorphisms (SNPs), deletions and insertions. Many SNPs have been identified and these data are now publicly available for researchers. For example, the International

HapMap Project (Helmuth, 2001), formed in 2002, aimed to characterize the patterns of linkage disequilibrium across the human genome using SNPs such that the information can be used for large-scale genetic association studies. As a dense SNP haplotype map is being built (Daly *et al.*, 2001; Helmuth, 2001; Patil *et al.*, 2001), various methods have been proposed to use haplotype information in linkage disequilibrium mapping. Some existing statistical methods for genetic linkage analysis have also shown increased power by incorporating SNP haplotype information (Huang *et al.*, 2004; Seltman *et al.*, 2001; Zhang *et al.*, 2002, 2003). But, the use of haplotype maps has been limited due to the fact that the human genome is a *diploid* and, in practice, *genotype* data instead of *haplotype* data are collected directly, especially in large-scale sequencing projects, because of cost considerations. Although recently developed experimental techniques (Douglas *et al.*, 2001) give the hope of deriving haplotype information directly with affordable costs, efficient and accurate computational methods for haplotype reconstruction from genotype data are still in high demand.

A number of methods have been developed to infer haplotypes based on genotypes of unrelated individuals. These methods can be divided into those based on combinatorics (Bafna *et al.*, 2003; Eskin and Halperin, 2003; Gusfield, 2001, 2002, 2003; Wang and Xu, 2003) and those based on expectation-maximization (EM) algorithms or bayesian algorithms (Excoffier and Slatin, 1995; Lin *et al.*, 2002; Niu *et al.*, 2002; Qin *et al.*, 2002; Stephens *et al.*, 2001, 2003). The statistical methods first infer haplotype frequencies and then use these frequencies to compute the haplotype configuration (or called *phase*) for each genotype. A recent study by Stephens and Donnelly (2003) compared three statistical approaches, the PL-EM algorithm (Niu *et al.*, 2002) called HAPLOTYPER, and two MCMC algorithms based on Gibbs sampling, one called PHASE (Stephens *et al.*, 2001) and another by Lin *et al.* (2002), using a variety of simulated and real genotype data. Two measures of accuracy were used: the error rate of individuals whose haplotype estimates are not completely correct, and the error rate of single site. The results showed that both error rates of these algorithms can be as high as 50%.

On the other hand, most combinatorics based methods consider two models. The first model is based on perfect phylogeny, assuming there is no recombination, and the other model is based

on pure parsimony, assuming the number of real haplotypes is minimum. In this paper, we study the pure parsimony model. Gusfield (2003) first formulated the problem and proposed an integer linear programming algorithm to solve this problem. Wang and Xu (2003) proposed a branching and bound algorithm called HAPAR to find the optimal solution. Recently, Brown and Harrower (2004) proposed a new formulation of the problem. Lancia *et al.* (2004) proved the APX-hardness of the problem. That is, if there is a constant $\lambda > 1$ such that the existence of a λ-approximation algorithm for this problem would imply P=NP. Sharan *et al.* (2005) showed that it remains APX-hard even in some very restricted cases.

In this paper, we first formulate the haplotype inference based on pure parsimony problem as an optimal haplotype inference (OHI) problem. Then the OHI problem is reformulated as an integer quadratic programming (IQP) problem. Based on the IQP problem, we propose an iterative semi-definite programming based approximation algorithm that finds a solution within a factor of $O(\log n)$ of the optimal solution, where n is the number of genotypes. We also prove that OHI is NP-hard through a reduction from the problem of Exact Cover By 3-Sets (X3C) (Garey and Johnson, 1979). This algorithm has been implemented and tested on a variety of simulated and biological data. In comparison with three other methods, HAPAR, HAPLOTYPER, and PHASE, the experimental results indicate that this algorithm outputs solutions with high accuracy and efficiency.

2 Method

Problem Formulation

Suppose we are given n individuals for a local chromosomal region of L linked SNPs. Let $G = \{g_1, g_2, ..., g_n\}$ denote the genotypes for the n individuals, where $g_i = \{g_{i1}, ..., g_{iL}\}$, g_{ij} denotes the genotype for individual i at locus j, and $g_{ij} = 0, 1$ or 2 denote that this locus is homozygous wild type, homozygous mutant, or heterozygous, respectively. Experimental data may have missing alleles. We let $g_{ij} = 3, 4$, or 5 to denote two missing alleles, one missing allele and one wild type, and one missing allele and one mutant.

Let $H = \{h_1, h_2, ..., h_m\}$ denote the set of all possible unobserved haplotypes for G. We denote $|H| = m$ to be the number of elements in a set. If two haplotypes h_r and h_t form a genotype g_i, we denote $h_r \otimes h_t = g_i$, and we also say that h_r and h_t resolve g_i, or a haplotype configuration for g_i is h_r and h_t. Let $S = \{S_1, ..., S_n\}$ denote the sets of unobserved haplotype configurations for G, where $S_i = \{(h_r, h_t) : h_r \otimes h_t = g_i\}$ denotes the set of all unobserved haplotype configurations for g_i. We formulate the haplotype inference by maximum parsimony as follows, which is referred to as optimal haplotype inference (OHI) problem.

> *Optimal Haplotype Inference(OHI)* Given a set of genotypes G and a polynomial-sized set of unobserved haplotypes H for G, ask to find a minimum subset of haplotypes, $V \subseteq H$, such that for every genotype g_i, $1 \leq i \leq n$, there exists a pair of haplotypes $h_r \in V$ and $h_t \in V$ such that h_r and h_t resolve g_i (or $(h_r, h_t) \in S_i$).

We would like to note that m (i.e., the size of H) is fixed in this paper because (1) the idea of haplotype inference is restricted to a high linkage disequilibrium (LD) region, which is usually short (Zhang *et al.*, 2002, 2004); and (2) the number of observed haplotypes in a short chromosomal region in human population is generally small. Theoretically, a genotype in a short chromosomal region may still contain a large number of ambiguous SNPs due to factors such as missing data, and thus corresponds to an exponential number of possible haplotypes. However, this kind of poor-quality genotypes do not provide enough information for haplotypes, so we do not use them for haplotype inference. If each genotype corresponds to a maximum number of K haplotypes, m is bounded by $O(nK)$.

Integer Quadratic Programming

Define x_i as the variable for haplotype h_i: $x_i = 1$ if $h_i \in V$, and $x_i = -1$ otherwise. Given a set of genotypes G, the OHI problem can be formulated as the following integer quadratic programming

problem,

$$IQP(\mathrm{G}): \quad \begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{m}(1+x_i)^2/4 \\ \text{subject to:} \quad & \sum_{(h_r,h_t)\in S_j}(1+x_r)(1+x_t)/4 \geq 1, \quad \forall j \in [1,n], \\ & x_i \in \{-1,1\}, \quad \forall i \in [1,m]. \end{aligned} \tag{1}$$

The set $V = \{i|x_i = 1\}$ corresponds to the set of selected haplotypes. The jth inequality guarantees that genotype $g_j \in G$ can be resolved.

Semidefinite Programming Relaxation

Since solving this integer quadratic programming is NP-complete, we consider relaxations of IQP. We can interpret IQP as restricting x_i to be a 1-dimensional vector with unit norm. Thus, we can relax x_i into a $(m+1)$-dimensional vector $\mathbf{y}_i$ of unit Euclidean norm. We introduce another $(m+1)$-dimensional unit vector $\mathbf{y}_0$, and relax IQP to

$$SDP(\mathrm{G}): \quad \begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{m}(\mathbf{y}_0+\mathbf{y}_i)^2/4 \\ \text{subject to:} \quad & \sum_{(h_r,h_t)\in S_j}(\mathbf{y}_0+\mathbf{y}_r)\cdot(\mathbf{y}_0+\mathbf{y}_t) \geq 4, \quad \forall j \in [1,n], \\ & |\mathbf{y}_i| = 1, \quad \forall i \in [1,m]. \end{aligned} \tag{2}$$

In fact, SDP becomes IQP if we let

$$\mathbf{y}_0 = (1,0,...,0), \mathbf{y}_1 = (x_1,0,...,0), ..., \mathbf{y}_m = (x_m,0,...,0). \tag{3}$$

SDP can be solved by semidefinite programming. Let $Y = (\mathbf{y}_0\mathbf{y}_1...\mathbf{y}_m)^T(\mathbf{y}_0\mathbf{y}_1...\mathbf{y}_m)$, where $y_{ij} = \mathbf{y}_i \cdot \mathbf{y}_j$. Then Y is positive semidefinite. We reformulate SDP into the following semidefinite programming:

$$\begin{aligned} \text{Minimize} \quad & C \cdot Y \\ \text{subject to:} \quad & A_j \cdot Y \geq a_j, \quad \forall j \in [1,n], \\ & y_{ii} = 1, \\ & Y \succeq 0, \end{aligned} \tag{4}$$

where $Y \succeq 0$ means Y is symmetric positive semidefinite. The semidefinite programming is an extension of the linear programming into convex cones. An efficient algorithm for the semidefinite programming is called the interior point method. Let OPT(SDP) be the optimal solution of SDP. For any given $\varepsilon > 0$, the interior point method finds a solution of value less than OPT(SDP)$+\varepsilon$ in time polynomial in the input size and $\log 1/\varepsilon$. Once an almost optimal solution Y is found, we can use an incomplete Cholesky decomposition to obtain vectors $\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_m$.

Algorithm SDPHapInfer

In the following, we introduce an algorithm that iteratively runs a semidefinite programming, finds a solution $\{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_m\}$, and constructs a solution $\{x_0, x_1, ..., x_m\}$ by *randomized rounding*.

Algorithm SDPHapInfer

1. Initialization
   (a) Let $U = G = \{g_1, ..., g_n\}$ be the set of unresolved genotypes;
   (b) Let $V = \{\}$ be the set of selected haplotypes;
2. SDP-Solving
   (a) Formulate IQP(U) and SDP(U);
   (b) Solve SDP(U), obtaining a solution $\{\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, ...\}$;
3. Randomized-Rounding
   (a) Randomly pick two multi-dimensional unit vectors $\mathbf{z}_1$ and $\mathbf{z}_2$;
   (b) Set $x_0 = 1$;
   (c) Set $x_i = 1$ for $i > 0$ if $(\mathbf{z}_1 \cdot \mathbf{y}_i)(\mathbf{z}_1 \cdot \mathbf{y}_0) > 0$ and $(\mathbf{z}_2 \cdot \mathbf{y}_i)(\mathbf{z}_2 \cdot \mathbf{y}_0) > 0$, $x_i = -1$ otherwise;
   (d) Let $V = V \cup \{h_i : x_i = 1\}$;
4. Iteration
   (a) Let U be the set of the genotypes that can not be resolved by V.

(b) If $|U| \neq 0$; goto Step 2;

5. Return V.

In Step 2, if a pair of haplotypes $h_r \in V$ and $h_t \notin V$ resolve $g_i \in U$, we set variable $\mathbf{y}_r = \mathbf{y}_0$ in SDP(U). Theoretically, we can run the SDP at Step 2 only once and use this result for randomized rounding for all the iterations without changing the time complexity, but practically, running the SDP for each iteration gives better solutions.

Analysis of Algorithm

Omitted

References

[1] Bafna, V., Gusfield, D., Lancia, G., and Yooseph, S. 2003. Haplotyping as perfect phylogeny: a direct approach. *J. Comp. Biol.*, 10:323–340.

[2] Brown, D., and Harrower I. 2004. A new integer programming formulation for the pure parsimony problem in haplotype analysis. *In Proc. WABI'04*, pages 254–265.

[3] Daly, M.J., Rioux, J.D., Schaffner, S.F., Hudson, T.J., and Lander, E.S. 2001. High-resolution haplotype structure in the human genome. *Nat. Genet.*, 29(2):229–232.

[4] Douglas, J.A., Boehnke, M., Gillanders, E., Trent, J.M., and Gruber, S.B. 2001. Experimentally-derived haplotypes substantially increase the efficiency of linkage disequilibrium studies. *Nat. Genet.*, 28(4):361–364.

[5] Drysdale, C., McGraw, D., Stack, C., Stephens, J., Judson, R., *et al*. 2000. Complex promoter and coding region β_2-adrenergic receptor haplotypes alter receptor expression and predict *in vivo* responsiveness. *Proc. Nat. Acad. Sci.*, 97:10483–10488.

[6] Eskin, E., and Halperin, E. 2003. Large scale recovery of haplotypes from genotype data using imperfect phylogeny. *In Proc. RECOMB'03*, pages 104–113.

[7] Excoffier, L., and Slatkin, M. 1995. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol.*, 12:921–927.

[8] Gabriel, S.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S.N., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E.S., Daly, M.J., and Altshuler, D. 2002. The structure of haplotype blocks in the human genome. *Science*, 296(5576):2225–2229.

[9] Garey, M.R., and Johnson, D.S. 1979. *Computers and intractability*, Freeman, New York.

[10] Gusfield, D. 2001. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *J. Comp. Biol.*, 8:305–323.

[11] Gusfield, D. 2002. Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. *In Proc. RECOMB'02*, pages 166–175.

[12] Gusfield, D. 2003. Haplotyping by pure parsimony. *In Proc. CPM'03, Lecture Notes in Computer Science*, 2676:144–155.

[13] Helmuth, L. 2001. Genome research: map of the human genome 3.0. *Science*, 293(5530):583–585.

[14] Huang, Y.-T., Zhang, K., Chen, T., and Chao, K.-M. 2004. Approximation algorithms for the selection of robust tag SNPs. *In Proc. WABI'04*, pages 278–289.

[15] Hudson, R.R. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18:337–338.

[16] Kerem, B., Rommens, J., Buchanan, J., Markiewicz, D., Cox, T., Chakravarti, A., Buchwald, M., and Tsui, L.C. 1989. Identification of the cystic fibrosis gene: genetic analysis. *Science*, 245:1073–1080.

[17] Lancia, G., Pinotti, C., and Rizzi., R. 2004. Haplotyping populations by pure parsimony: complexity of exact and approximation algorithms. *INFORMS J. Comp.*, 16:348–359.

[18] Lin, S., Cutler, D.J., Zwick, M.E., and Chakravarti, A. 2002. Haplotype inference in random population samples. *Am. J. Hum. Genet.*, 71:1129–1137.

[19] Niu, T., Qin, Z., Xu, X., and Liu, J.S. 2002. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am. J. Hum. Genet.*, 70:157–159.

[20] Patil, N., Berno, A.J., Hinds, D.A., Barrett, W.A., Doshi, J.M., Hacker, C.R., Kautzer, C.R., Lee, D.H., Marjoribanks, C., McDonough, D.P., *et al*. 2001. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723.

[21] Qin, Z., Niu, T., and Liu, J. 2002. Partitioning-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide Ploymorphisms. *Am. J. Hum. Genet.*, 71:1242–1247.

[22] Seltman, H., Roeder, K., and Devlin, B. 2001. Transmission/disequilibrium test meets measured haplotype analysis: family-based association analysis guided by evolution of haplotypes. *Am. J. Hum. Genet.*, 68(5):1250–1263.

[23] Sharan, R., Halldorsson, B.V., and Istrail, S. 2005. Islands of Tractability for Parsimony Haplotyping. *To appear in Proc. CSB'05.*

[24] Stephens, M., Smith, N.J., and Donnelly, P. 2001. A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.*, 68(4):978–989.

[25] Stephens, M., and Donnelly, P. 2003. A comparison of bayesian methods for haplotype reconstruction from population genotype data. *Am. J. Hum. Genet.*, 73:1162–1169.

[26] Wang, L., and Xu, Y. 2003. Haplotype inference by maximum parsimony. *Bioinformatics*, 19(14):1773–1780.

[27] Zhang, K., Deng, M., Chen, T., Waterman, M.S., and Sun, F. 2002. A dynamic programming algorithm for haplotype partitioning. *Proc. Nat. Acad. Sci.*, 99(11):7335–7339.

[28] Zhang, K., Sun, F., Waterman, M.S., and Chen, T. 2003. Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data. *Am. J. Hum. Genet.*, 73:63–73.

[29] Zhang, K., Qin, Z.S., Liu, J.S., Chen, T., Waterman, M.S., and Sun, F. 2004. Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies. *Genome Res.*, 14(5):908–916.