Approximation Algorithms for the Optimization Problems of SNPs and Haplotypes

Yao-Ting Huang
Department of Computer Science
and Information Engineering
National Taiwan University, Taipei, Taiwan
Email: ythuang@acb.csie.ntu.edu.tw

Kun-Mao Chao[1,2]
[1]Department of Computer Sciecne
and Information Engineering
[2]Graduate Institute of Networking and Multimedia
National Taiwan University, Taipei, Taiwan
Email: kmchao@csie.ntu.edu.tw

***Abstract*— This paper studies two optimization problems in the SNP and haplotype research. The first problem asks for a minimum set of SNPs that can tolerate a certain number of missing data. The second problem asks for a minimum set of haplotypes that can explain a given set of genotypes. We show that both problems are NP-hard and design several approximation algorithms to solve them efficiently. These algorithms have been implemented and tested on both simulated and biological data. Our theoretical analysis and experimental results indicate that these algorithms are able to find solutions close to the optimal solutions.**

I. Introduction

The next important step in human genomics is to correlate genetic variations with phenotypic differences. In the human genome, the common genetic variations include insertions, deletions, and *Single Nucleotide Polymorphisms* (SNPs). SNPs have become the preferred markers for association studies of genetic diseases and traits because of their high abundance and available high throughput genotyping technologies. A SNP is a single DNA base mutation which is kept through heredity. The value of a SNP is usually assumed to be binary since it's either the major or the mutant allele. A set of linked SNPs on one chromosome is called a *haplotype*.

Recent findings have shown that the human haplotypes have a block-like structure, and a small subset of SNPs in the block (called "tag SNPs") is sufficient to distinguish each pair of patterns in the block [1], [13], [16], [17], [18]. However, a SNP may be genotyped as missing data (i.e., we fail to obtain the value of the SNP) if it does not pass the threshold of data quality [13], [18], [19]. If only using the minimum set of tag SNPs, we may fail to distinguish a haplotype sample when some tag SNPs are missing. Under the current genotyping technology, the occurrence of missing data still can not be avoided. In order to distinguish a haplotype sample unambiguously, we have to work on a set of SNPs which is not affected by missing data.

Moreover, the use of haplotype information has been limited due to the fact that the human genome is a diploid (i.e., the chromosomes appear in pairs). A pair of haplotypes are called a *genotype*. In large-scale sequencing projects, genotype data instead of haplotype data are obtained directly due to cost considerations [12], [14], [15]. Based on genotype data, we can only know the two alleles at each SNP locus of a pair of haplotypes. However, we lose the information which haplotype the allele appears in. In order to obtain more accurate genetic information for association studies, we have to know not only the genotype data but also the haplotype data. As a result, efficient and accurate computational methods for inferring haplotype data from genotype data are still highly demanded.

In this paper, we summarize our previous results for solving the above two problems [7], [8]. To solve the problem of missing data, we show there exists a set of SNPs, called robust tag SNPs, which is able to tolerate a certain number of missing SNPs [7]. Several algorithms are introduced to find the minimum set of robust tag SNPs. To infer haplotype data from genotype data, we propose an iterative semi-definite programming relaxation algorithm [8]. This algorithm searches for a minimum set of haplotypes that can explain a given set of genotypes. All developed algorithms have been implemented and tested using a variety of simulated and biological data. The experimental results and theoretical analysis show that our algorithms is not only efficient but also able to find a solution close to the optimal solution..

II. Finding Robust Tag SNPs

Assume we are given a haplotype block containing N SNPs and K haplotype patterns. Let m be the maximal number of missing SNPs we wish to tolerate. This problem asks for a minimum set of robust tag SNPs in the haplotype block which is able to tolerate up to m missing SNPs. The problem of finding the minimum set of robust tag SNPs is shown to be NP-hard. To find robust tag SNPs efficiently, we design and implement two greedy and one iterative linear-programming (LP) relaxation algorithms. For the details of these approximation algorithms, please refer to [7].

- The first and second greedy algorithms select the robust tag SNPs one by one in different greedy manners. Both of these algorithms have been implemented in Java. These algorithms run in polynomial time and are able to find a solution within a factor of $(m+1)\ln\frac{K(K-1)}{2}$ and $\ln((m+1)\frac{K(K-1)}{2})$ of the optimal solution respectively.
- We formulate this problem as an integer linear programming problem and design an iterative LP-relaxation algorithm. The iterative LP-relaxation algorithm has been implemented in Perl, and is able to find a solution within

0-7803-9329-5/05/$20.00 ©2005 IEEE

a factor of $O(m \ln K)$ of the optimal solution. Since the iterative LP-relaxation algorithm is a randomized method, this program is repeated for 10 times and the best solution among them is chosen as the output.

We have implemented and tested these algorithms on a variety of simulated and biological data, including randomly generated haplotype data, Hudson's simulated data [9], Daly's Chromosome 5q31 data [3], and Patil's Chromosome 21 data [13]. In addition, we also implement a Java program which enumerates all possible solutions to search the optimal solution.

1. The experimental result shows that the optimal solution is not computable as the size of the data set and the value of m increase. On the other hand, all of our programs are able to output a solution in seconds. In the experiments where the optimal solution still can be found, the solutions returned by our approximation algorithms are quite close to the optimal solution.
2. The genotyping cost saved by using tag SNPs ranges from 75% to 90% in different data sets. In addition, the genotyping cost of extra tag SNPs for tolerating missing data is still less than 50% even when m increases to 6. Therefore, the result indicates that genotyping robust tag SNPs for tolerating missing data is still cost-effective.

III. Inferring Haplotypes from Genotypes

Given a set of n genotypes and a set of m possible haplotypes, we wish to find a minimum subset of haplotypes such that each genotype can be resolved by two haplotypes from this subset. This problem has been extensively studied and shown to be APX-hard [11], [12], [14], [15]. We formulate this problem as an integer quadratic programming (IQP) problem and propose an iterative semi-definite programming relaxation algorithm (called SDPHapInfer). For the detail of this algorithm, please refer to [8].

- The SDPHapInfer algorithm iteratively runs the following steps: (1) formulates the problem as an IQP problem, (2) relaxes the integer constraint and reformulates IQP to a semi-definite programming (SDP) problem, (3) solves the SDP problem, and (4) constructs the integer solution by a randomized rounding method based on the SDP solution. This process is repeated until all of the constraints in the IQP problem are satisfied. We prove that SDPHapInfer is able to return a solution within a factor of $O(\log n)$ of the optimal solution.

SDPHapInfer has been implemented in MatLab and tested on a variety of simulated and biological data, including randomly generated haplotype data, Hudson's simulated data [9], β_2-adrenergic receptors gene [10] and cystic fibrosis gene [4]. The experimental results of SDPHapInfer are compared with a branching and bound algorithm called HAPAR [15], a PL-EM algorithm called HAPLOTYPER [12], and PHASE [14] which combined the Gibbs sampling algorithm with an approximate coalescent prior.

1. The experimental results indicate that the number of haplotypes found by PHASE is significantly larger than the others on random data sets. It is because PHASE incorporates the coalescent prior in their algorithm which does not fit these data sets.
2. We also evaluate these programs by a common criterion, error rates, in the haplotype inference study. The results shows that SDPHapInfer and HAPLOTYPER have similar error rates. In addition, the results generated by PHASE have lower error rates on some data sets but obtain higher error rates on others. The error rates of HAPAR are higher than the others on biological data sets.
3. In terms of efficiency, HAPAR is the fast one when the size of the data set is small. On the other hand, SDPHapInfer, HAPLOTYPER, and PHASE always output a solution in a stable and consistent way, and they run much faster than HAPAR when the number of genotypes become large.

IV. Conclusion

This paper studies two optimization problems related to SNP and haplotype. To avoid the influence of missing data, we show there exists a subset of robust tag SNPs which is able to tolerate a certain number of missing data. Several approximation algorithms are designed and implemented to find the minimum set of robust tag SNPs. For the haplotype inference problem, we propose an iterative SDP-relaxation algorithm to find a minimum set of haplotypes that can explain a given set of genotypes. Our theoretical analysis and experimental results both show that our algorithms are not only efficient but also able to find solutions close to the optimal solution.

Acknowledgment

Yao-Ting Huang and Kun-Mao Chao were supported in part by an NSC grant 93-2213-E-002-029.

References

[1] Bafna, V., Halldórsson, B.V., Schwartz, R., Clark, A.G., and Istrail, S. Haplotypes and informative SNP selection algorithms: don't block out information. *In Proc. RECOMB'03*, pages 19–27, 2003.

[2] Cormen T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. *Introduction to algorithms*, The MIT Press, 2001.

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

[4] Drysdale, C., McGraw, D., Stack, C., Stephens, J., Judson, R., *et al*. 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, 2000.

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

[6] Halldórsson, B.V., Bafna, V., Lippert, R., Schwartz, R., Vega, F.M., Clark, A.G., and Istrail, S. Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies. *Genome Research*, pages 1633–1640, 2004.

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

[8] Huang, Y.-T., Chao, K.-M., and Chen, T. An approximation algorithm for haplotype inference by pure parsimony. *To appear in Journal of Computational Biology*, 2005.

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

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

[11] Lancia, G., Pinotti, C., and Rizzi., R. 2004. Haplotyping Populations: complexity, exact and approximation algorithms. To appear in INFORMS J. Comp. 2004.

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

[13] 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*. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

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

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

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

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

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

[19] Zhao, J.H., Lissarrague, S., Essioux, L., and Sham, P.C. GENECOUNTING: haplotype analysis with missing genotypes. *Bioinformatics*, 18:1694–1695, 2002.