Perfect Phylogeny Haplotyper: Haplotype Inferral Using a Tree Model

Ren Hua Chung and Dan Gusfield*

September 9, 2002

Abstract

Summary: We have developed an efficient program, the Perfect Phylogeny Haplotyper (PPH), that takes in unphased population genotype data, and determines if that data can be explained by haplotype pairs that could have evolved on a perfect phylogeny. **Availability:** Executable code for four common platforms is available at: wwwcsif.cs.ucdavis.edu/~gusfield **Contact** gusfield@cs.ucdavis.edu

1 Introduction

Programs for inferring haplotype pairs from population genotype data are needed in many haplotyping efforts and will be increasingly valuable in combination with the proposed NIH Haplotype Mapping Project. Successful haplotype inferral by computer requires a genetic model of haplotypes in a population. The *perfect phylogeny* model is a strong model that is justified by recent molecular observations. Program PPH determines whether unphased genotypes in a population can be explained by haplotype pairs that fit the perfect phylogeny model.

In diploid organisms (such as humans) there are two (not completely identical) "copies" of each chromosome, and hence of each region of interest. A description of the data from a single copy is called a *haplotype*, while a description of the conflated (mixed) data on the two copies is called a *genotype*. In complex diseases (those affected by more than a single gene) it is often much more informative to know the haplotypes (identifying a set of gene alleles inherited together) than to only know the genotypes. However, it is expensive or technically difficult to examine the two copies of a chromosome separately, and so *genotypes* rather than haplotypes are obtained. Then one tries to computationally infer the haplotypes from the genotypes. That goal would be impossible without the implicit or explicit use of some genetic model, either to assess the biological fidelity of any proposed solution, or to guide the algorithm in constructing a solution. In [6], we considered the genetic model where the inferred haplotypes are required to fit a *perfect phylogeny*, defined below.

Genotype data is represented as an n by m 0-1-2 (ternary) matrix G. Each row is a genotype. A pair of binary vectors of length m (haplotypes) explain a row i of G if for every position c both entries in the haplotypes are 0 (or 1) if and only if $G(i,c)$ is 0 (or 1) respectively, and exactly one entry is 1 and one is 0 if and only if $G(i,c) = 2$.

Let M be an $2n$ by m 0-1 (binary) matrix, and let V be an m-entry binary matrix. A *perfect phylogeny for M and V* is a rooted tree T with exactly $2n$ leaves, where each leaf is labeled by one of the rows of M, and each row labels one leaf. Each column labels exactly one edge of T. For any row i, the labels on the edges on the path from the root of T to the leaf labeled i, specify exactly those positions where an entry in row i differs from the corresponding entry in V. Hence, that path, along with V, is a representation of row i, and T with V is a compact representation of M. Tree T is

*Dept. of Computer Science, 3051 Engineering II, University of California, One Shields Avenue, Davis, CA 95616. Email: gusfield@cs.ucdavis.edu Research Supported by NSF grant DBI-9723346

a possible explanation for the evolution of the rows of M, under the assumption that each mutation (change from 0 to 1 in a position) happens exactly once. The justification for the perfect phylogeny model is based on recent observations of little or no recombination in long segments of DNA [3, 8, 4], and the standard infinite-sites assumption of population genetics. See [9, 6] for a more complete justification of this model.

There is also a version of perfect phylogeny where no vector V is specified. Then, a binary matrix M is said to have a perfect phylogeny if *there exists* a V such that there is a perfect phylogeny for M and V.

Finally, the *Perfect Phylogeny Haplotype (PPH) Problem* is: Given an n by m ternary matrix of genotypes G, produce a $2n$ by m binary matrix M and a vector V, so that: 1) for each row i of G, rows $2i-1$ and $2i$ of M are haplotypes that explain genotype i and 2) there is a perfect phylogeny for M and V.

The solution to the PPH problem given in [6] is based on reducing it to a well-studied problem in graph theory, called the *Graph Realization Problem.* See wwwcsif.cs.ucdavis.edu/˜gusfield/recomberrata.pdf for a simpler reduction that also fixes a small problem in the paper. Program PPH implements the details of that reduction. After reducing a problem instance, the PPH program solves the graph realization instance and translates the solution back to the PPH problem. There are several efficient solutions known to the graph realization problem. The method in [2] runs in $O(nm\alpha(nm))$ time, where α is the inverse-Ackerman function, which grows so slowly that it is taken to be a constant term. Several other methods, which are simpler to program, solve the graph realization problem in $O(n^2m)$ time or $O(nm^2)$ time. In the program PPH, we have used one of the later methods, by Gavril and Tamari [5], which is a variant of Tutte's classic algorithm for graph realization [10].

The PPH program solves three variants of the PPH problem: when V is explicitly specified, when no V is given; and when V is assumed to be the all-0 vector. The package is mostly written in C++, but relies on some procedures in Perl. Hence a Perl interpreter is needed, even when running the compiled C++ code. The reduction phase of the PPH program takes $O(nm)$ time, and so the program has a theoretical worst case running time of $O(nm^2)$. In addition to the proof of correctness of the algorithm, we tested the program using haplotype data generated by an early version of R. Hudson's program ms [7]. Pairs of haplotypes generated by Hudson's program were randomly paired in order to create genotype data. The program was tested on tens of thousands of examples and proved correct in all cases. The program verifies that the haplotypes generated explain the genotypes, and it also outputs the corresponding perfect phylogeny in New Hampshire tree format. The program also determines whether the solution is unique or whether there is a different solution to the same genotype data.

For genotype data with 100 individuals (n) and 100 sites (m), the program typically solves the problem in under one second on a Powerbook G4 computer. Since the PPH program reduces the PPH problem to a problem of graph realization, we separated the code into several procedures, one for the reduction (about one thousand lines of C++), and one to solve the general graph realization problem with arbitrary input (about four thousand lines of C++). Hence this package is useful not only for solving PPH problem but for other applications of graph realization.

In [6], we also show how to represent all the solutions to the PPH problem when the solution is not unique. That algorithm takes linear time once a first solution is known. We have not implemented that algorithm in the PPH package, but that functionality should be available shortly through a different algorithm which is presently being implemented [1].

Compiled versions for Sun Solaris, Red Hat Linux 7.1, MS Windows 2000, and MAC Powerbook G4 are available at: wwwcsif.cs.ucdavis.edu/˜gusfield

References

[1] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. Technical report, UC Davis, Department of Computer Science, 2002.

[2] R. E. Bixby and D. K. Wagner. An almost linear-time algorithm for graph realization. *Mathematics of Operations Research*, 13:99–123, 1988.

[3] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29:229–232, 2001.

[4] L. Friss, R. Hudson, A. Bartoszewicz, J. Wall, T. Donfalk, and A. Di Rienzo. Gene conversion and differential population histories may explain the contrast between polymorphism and linkage disequilibrium levels. *Am. J. of Human Genetics*, 69:831–843, 2001.

[5] F. Gavril and R. Tamari. An algorithm for constructing edge-trees from hypergraphs. *Networks*, 13:377–388, 1983.

[6] D. Gusfield. Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions (Extended Abstract). In *Proceedings of RECOMB 2002: The Sixth Annual International Conference on Computational Biology*, pages 166–175, 2002.

[7] R. Hudson. Generating samples under the Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

[8] J. C. Stephens and et. al. Haplotype variation and linkage disequilibrium in 313 human genes. *Science*, 293:489–493, 2001.

[9] S. Tavare. Calibrating the clock: Using stochastic processes to measure the rate of evolution. In E. Lander and M. Waterman, editors, *Calculating the Secretes of Life*. National Academy Press, 1995.

[10] W.T. Tutte. An algorithm for determining whether a given binary matroid is graphic. *Proc. of Amer. Math. Soc*, 11:905–917, 1960.