ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"

CONSIGLIO NAZIONALE DELLE RICERCHE

L. Tininini, P. Bertolazzi, A. Godi

COLLHAPS: A NEW HEURISTIC ALGORITHM FOR HAPLOTYPE INFERENCE BY MAXIMUM PARSIMONY

R. 616 Ottobre 2004

Leonardo Tininini – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: `tininini@iasi.rm.cnr.it`.

Paola Bertolazzi – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: `bertola@iasi.rm.cnr.it`.

Alessandra Godi – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: `godi@iasi.rm.cnr.it`.

The authors wish to thank Giuseppe Lancia for many valuable discussions on the topic and the authors of the programs used in the experiments for providing their software.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

In this paper we propose a new algorithm for haplotype inference by maximum parsimony, based on the consecutive application of a generalized version of the well-known Clark's rule, called collapse rule. The algorithm has been implemented in a software called CollHaps and tested on several real and simulated datasets, demonstrating that it enables the user to process large datasets in short processing times without sacrificing solution accuracy.

1. Introduction

Recent work in genome sequencing has shown that all humans share about 99.9% of their DNA [Ven+01], the main differences lying in nucleotide variants at specific base positions, commonly known as SNPs (Single Nucleotide Polymorphisms) [CoBC98]. The variants at SNP sites are called *alleles* and, for reasons still unclear, SNPs are almost always biallelic, i.e. only two of the four possible variants have a non-negligible frequency (1% or greater). The exact sequence of alleles along each of the two chromosome copies in diploid organisms (e.g. humans) is called a *haplotype* and each SNP site is called either *homozygous* or *heterozygous*, depending on whether the alleles are the same or different on the two haplotypes.

The determination of individual haplotypes is fundamental in many practical contexts, e.g. detecting diseases and drug design, as well as in evolutionary studies on populations [JUD+99, HKW+00]. Unfortunately, current routine sequencing technology can only determine conflated sequences, known as *genotypes*, where information on which alleles came from the same chromosome copy is unknown for heterozygous sites. Experimental techniques to obtain individual haplotypes are very expensive, time consuming and labor intensive. This motivates the increasing interest in computational techniques for *haplotype inference*, i.e. aimed at determining haplotype pairs from the corresponding genotypes [Gu04, HBE+04, LaPR04].

In [Cl90] Clark proposed a haplotype inference technique based on the iterative application of a well-known inference rule. Some algorithms are based on expectation-maximization (EM): starting from an initial guess, the haplotype frequencies are iteratively updated, trying to maximize a likelihood function [HaKi95, LWU95]. More recently, [KiSh05] proposed a new EM-based algorithm for the simultaneous identification of haplotypes and SNP blocks. Other approaches are based on Markov Chain Monte Carlo methods [NQXL02, StDo03]. A further category is that of algorithms based on the *maximum parsimony principle*, whose aim is to minimize the number of distinct haplotypes used to explain the given set of genotypes. A branch and bound algorithm based on maximum parsimony was proposed in [WaXu03], while Integer Linear Programming techniques were proposed in [Gu03, BrHa04]. The maximum parsimony principle has been justified by both experimental results and theoretical arguments [Gu03, WaXu03].

In this paper we introduce a generalization of the above mentioned Clark's inference rule, called *collapse rule*, and show some relevant properties related to it. We also present a new heuristic algorithm for haplotype inference, based on the maximum parsimony principle and on the consecutive application of collapse rules. Finally, the performance of the proposed algorithm is tested on several real and simulated datasets, demonstrating its scalability properties, enabling the user to process large datasets in short processing times without sacrificing the accuracy of the solution.

2. The Haplotype Inference Problem

In this section we formally introduce the Haplotype Inference Problem (HIP) and the measures used in Section 4 to compare algorithm performance.

2.1. A formalization of the problem

As SNPs in humans are almost always biallelic, the two alleles at any SNP site are commonly encoded by the symbols 0 and 1, independently of the actual bases constituting the two variants. A haplotype h^* with m SNPs is therefore represented by a m-dimensional vector, such that each

component $h^*_j \in \{0,1\}$. Similarly, a genotype g is represented by a m-dimensional vector, where each component $g_j \in \{0,1,2\}$: 0 and 1 relate to homozygous sites, while heterozygous sites are denoted by 2. We introduce the conflate operator $\oplus : \{0,1\} \to \{0,1,2\}$, defined as follows:

$$\begin{cases} 0 \oplus 0 = 0 \\ 0 \oplus 1 = 1 \oplus 0 = 2 \\ 1 \oplus 1 = 1 \end{cases}$$

which generalizes to vectors in the obvious way: given a m-dimensional genotype g and a pair of m-dimensional haplotypes h^*_1 and h^*_2,

$$g = h^*_1 \oplus h^*_2 \iff g_j = h^*_{1,j} \oplus h^*_{2,j} \ (j = 1, \ldots, m)$$

An instance of HIP (*HIP-instance*) is constituted by an $n \times m$ matrix G such that each row g_i is a m-dimensional genotype. A corresponding (candidate) solution is a $2n \times m$ matrix H^* such that each row h^*_i is a m-dimensional haplotype and:

$$g_i = h^*_{2i-1} \oplus h^*_{2i} \ (i = 1, \ldots, n)$$

We call the pair of haplotypes h^*_{2i-1} and h^*_{2i} the *genotype solution* of g_i. We are interested in solutions satisfying the above *maximum parsimony principle*: one of the possible candidate solutions is searched, such that the number of distinct haplotypes in $\{h^*_1, h^*_2, \ldots, h^*_{2n}\}$ is minimum.

Let us consider a HIP-instance and associate a distinct variable x_p to each '2' in it. For each genotype g_i two *symbolic haplotypes* h_{2i-1} and h_{2i} are derived, defined by:

$$h_{2i-1,j} = \begin{cases} 0 & \text{if } g_{i,j} = 0 \\ 1 & \text{if } g_{i,j} = 1 \\ x_p & \text{if } g_{i,j} = 2 \end{cases} \qquad h_{2i,j} = \begin{cases} 0 & \text{if } g_{i,j} = 0 \\ 1 & \text{if } g_{i,j} = 1 \\ \bar{x}_p & \text{if } g_{i,j} = 2 \end{cases}$$

where the variables x_p take values from $\{0,1\}$ and $\bar{x}_p$ is a shorthand for $1 - x_p$ (obviously $\bar{\bar{x}}_p = x_p$). The variables x_p and $\bar{x}_p$ are called *complementary* and the $2n \times m$ matrix H, whose rows are the symbolic haplotypes h_i, is called *symbolic solution*. According to the maximum parsimony principle we want to determine a variable assignment for the variables x_p such that the resulting number of distinct haplotypes is minimum.

Note that variable assignments induce a partition among the symbolic haplotypes h_1, h_2, $\ldots$, h_{2n}: two symbolic haplotypes h_p and h_q are in the same partition class iff the variable assignment maps them to the same haplotype h^*. Hence a solution is optimal (with respect to the maximum parsimony principle) iff the number of classes is minimum.

Example 1. *Given the HIP-instance constituted by the three genotypes: 1022, 2220, 2202, a symbolic solution is given by the three pairs of symbolic haplotpyes:*

$$\begin{array}{lcccc} h_1: & 1 & 0 & x_1 & x_2 \\ h_2: & 1 & 0 & \bar{x}_1 & \bar{x}_2 \\ h_3: & x_3 & x_4 & x_5 & 0 \\ h_4: & \bar{x}_3 & \bar{x}_4 & \bar{x}_5 & 0 \\ h_5: & x_6 & x_7 & 0 & x_8 \\ h_6: & \bar{x}_6 & \bar{x}_7 & 0 & \bar{x}_8 \end{array}$$

A candidate solution is an assignment for the variables $x_1, \ldots, x_8$. In particular, the following variable assignment: x_1=0, x_2=0, x_3=1, x_4=0, x_5=0, x_6=1, x_7=0, x_8=0 corresponds to a

candidate solution with 4 haplotypes, namely 1000 $\{h_1^*, h_3^*, h_5^*\}$*, 1001* $\{h_2^*\}$*, 0110* $\{h_4^*\}$ *and 0101* $\{h_6^*\}$*:*

$$g_1 = 1022 = h_1^* \oplus h_2^* = 1000 \oplus 1011$$
$$g_2 = 2220 = h_3^* \oplus h_4^* = 1000 \oplus 0110$$
$$g_3 = 2202 = h_5^* \oplus h_6^* = 1000 \oplus 0101$$

while the variable assignment: x_1*=0,* x_2*=1,* x_3*=1,* x_4*=0,* x_5*=1,* x_6*=0,* x_7*=1,* x_8*=0 corresponds to an (optimal) solution with 3 haplotypes, namely 1001* $\{h_1^*, h_6^*\}$*, 1010* $\{h_2^*, h_3^*\}$ *and 0100* $\{h_4^*, h_5^*\}$*:*

$$g_1 = 1022 = h_1^* \oplus h_2^* = 1001 \oplus 1010$$
$$g_2 = 2220 = h_3^* \oplus h_4^* = 1010 \oplus 0100$$
$$g_3 = 2202 = h_5^* \oplus h_6^* = 0100 \oplus 1001$$

2.2. Performance measures

To compare the performance of the proposed algorithm with respect to others, the following measures will be considered:

- *genotype error rate.* Very commonly used, it is defined as the percentage of incorrectly inferred genotypes. In this paper, however, we will normally use the switch error rate, as it is a more precise measure of the algorithm accuracy.

- *switch error rate.* This is a more accurate version of the previous measure and has already been used in [LCZC02, StDo03]. It is based on the (minimum) number of switches needed on the heterozygous sites of a solution to recover the original correct haplotypes. It is easily seen that given a genotype g_i with s_i heterozygous sites, this value can be at most $\lfloor \frac{s_i}{2} \rfloor$. In the case of a HIP-instance comprising n genotypes, the maximum (worst-case) total number of switches required to recover the original haplotypes is therefore expressed by

  $$\sum_{i=1}^{n} \left\lfloor \frac{s_i}{2} \right\rfloor$$

  The switch error rate is defined as the ratio between the total number of switches measured on the inferred solution and the maximum total number of switches expressed above.

- *number of haplotypes used.* This value measures the algorithm compliance to the maximum parsimony principle: good algorithms should use fewer haplotypes in the inferred solutions.

- *processing time and size constraints.* The time required to produce the solution and explicit or implicit size constraints are obviously indicators of the algorithm's scalability and practical applicability. Even for medium-scale problem instances, some programs may require hours or days of processing time, or even crash due to memory allocation problems.

3. The CollHaps Algorithm

In this section a new haplotpye inference algorithm based on the maximum parsimony principle will be illustrated. It is based on the collapse rule concept (a generalization of Clark's rule) and on a sophisticated technique for the solution's progressive improvement.

3.1. The collapse rule

A very simple, well-known technique for HIP is based on the iterative application of "Clark's rule": this can be applied to a genotype g and a "known" compatible haplotype h_1^* (stemming from unambiguous genotypes or a previous rule application) to infer a new "known" haplotype h_2^* such that: $g = h_1^* \oplus h_2^*$. In this section we introduce the concept of collapse rule and show that it generalizes Clark's rule to symbolic haplotypes. A collapse rule corresponds to the minimum set of variable assignments, which forces the equality of two symbolic haplotypes. A prerequisite for the application of a collapse rule to a pair of symbolic haplotypes h' and h'' is their compatibility.

Definition 1. *(compatible symbolic haplotypes) Two k-dimensional symbolic haplotypes h' and h'' are compatible (for collapse) iff for each $j \in \{1, \ldots, k\}$ one of the following holds:*

- $h'_j = h''_j = 0$
- $h'_j = h''_j = 1$
- *either h'_j or h''_j is a variable (but not both)*
- *both h'_j and h''_j are variables and not complementary*

Definition 2. *(collapse assignment) Given two compatible symbolic haplotypes h', h'' the collapse assignment (for h', h'') is the variable assignment ϑ defined as follows:*

- *if $h'_j = x_p$ and h''_j is a constant $c \in \{0/1\}$ then $\vartheta(x_p) = c$*
- *if $h'_j = \bar{x}_p$ and h''_j is a constant c (0/1) then $\vartheta(x_p) = 1 - c$*
- *if h'_j is a constant c (0/1) and $h''_j = x_q$ then $\vartheta(x_q) = c$*
- *if h'_j is a constant c (0/1) and $h''_j = \bar{x}_q$ then $\vartheta(x_q) = 1 - c$*
- *if $h'_j = x_p$ and $h''_j = x_q$ then $\vartheta(x_q) = x_p$*
- *if $h'_j = \bar{x}_p$ and $h''_j = \bar{x}_q$ then $\vartheta(x_q) = x_p$*
- *if $h'_j = x_p$ and $h''_j = \bar{x}_q$ then $\vartheta(x_q) = \bar{x}_p$*
- *if $h'_j = \bar{x}_p$ and $h''_j = x_q$ then $\vartheta(x_q) = \bar{x}_p$*
- *ϑ is the identity for any other variable*

Definition 3. *(application of a collapse rule on a matrix of haplotypes) Given a matrix of symbolic haplotypes H and a pair of compatible symbolic haplotypes $h', h'' \in H$, the application of a collapse rule for h', h'' on H is the matrix obtained by applying the collapse assignment for h', h'' on all symbolic haplotypes in H.*

It is evident that Clark's rule is a particular form of collapse rule, corresponding to the case where one of the two haplotypes to be collapsed is unambiguous (i.e. does not contain variables). As noted above, the application of a collapse assignment makes the symbolic haplotypes h' and h'' equal and hence after the application of a collapse rule the number of distinct elements in the list H is decreased by (at least) one. The following proposition states a fundamental property of the collapse rule.

Proposition 3.1. *Given one optimal (in terms of parsimony) solution for a HIP-instance, there always exists a collapse rule application sequence, which produces a set of optimal solutions including at least the given one.*

Proof. (sketch) The construction of the sequence is trivial: we have seen that a HIP solution is a variable assignment for the variables in the symbolic haplotypes which defines a partition on the list of symbolic haplotypes. For each of the N partition classes defined by the given optimal solution, we apply a sequence of collapse steps on the member haplotypes in order to make them all equal (if the class has cardinality n_c this is performed in $n_c - 1$ steps). The sequence of collapse steps will yield exactly N distinct (possibly still partially symbolic) haplotypes. Any solution obtained by arbitrarily assigning 0/1 values to the residual variables is optimal and one of them is obviously the given one. ∎

Example 2. *Given the HIP-instance of Example 1 and its optimal solution, one possible sequence of collapse rules (the order by which the rule is applied to the single pairs is unimportant) is: (h_1, h_6), (h_2, h_3), (h_4, h_5). The initial matrix of symbolic haplotypes is the following:*

$$\begin{array}{lcccc} h_1: & 1 & 0 & x_1 & x_2 \\ h_2: & 1 & 0 & \bar{x}_1 & \bar{x}_2 \\ h_3: & x_3 & x_4 & x_5 & 0 \\ h_4: & \bar{x}_3 & \bar{x}_4 & \bar{x}_5 & 0 \\ h_5: & x_6 & x_7 & 0 & x_8 \\ h_6: & \bar{x}_6 & \bar{x}_7 & 0 & \bar{x}_8 \end{array}$$

The first application of the collapse rule (for h_1, h_6) corresponds to the following variable assignments: x_1=0, x_6=0, x_7=1, x_8=$\bar{x}_2$ and produces the following matrix:

$$\begin{array}{lcccc} h_1: & 1 & 0 & 0 & x_2 \\ h_2: & 1 & 0 & 1 & \bar{x}_2 \\ h_3: & x_3 & x_4 & x_5 & 0 \\ h_4: & \bar{x}_3 & \bar{x}_4 & \bar{x}_5 & 0 \\ h_5: & 0 & 1 & 0 & \bar{x}_2 \\ h_6: & 1 & 0 & 0 & x_2 \end{array}$$

The second application (for h_2, h_3) corresponds to the variable assignment: x_2=1, x_3=1, x_4=0, x_5=1 and produces the following matrix:

$$\begin{array}{lcccc} h_1: & 1 & 0 & 0 & 1 \\ h_2: & 1 & 0 & 1 & 0 \\ h_3: & 1 & 0 & 1 & 0 \\ h_4: & 0 & 1 & 0 & 0 \\ h_5: & 0 & 1 & 0 & 0 \\ h_6: & 1 & 0 & 0 & 1 \end{array}$$

The final application (to h_4, h_5) is not necessary, as h_4 and h_5 have already been made identical by the previous assignments.

Proposition 3.1 is of theoretical interest, but does not provide a practical strategy to obtain the optimal sequence of collapse steps. Since the exhaustive exploration of all collapse sequences is practically unfeasible, a sophisticated heuristic strategy has been elaborated and implemented by the authors in a program called CollHaps and is described below.

3.2. The preprocessing

Like most programs for HIP, CollHaps performs an initial preprocessing to try to reduce the number of symbolic haplotypes on which the collapse rules are to be applied. This is obtained by removing duplicate genotypes, and is transparent to the user.

The CollHaps algorithm

for i = 0 **to** maxExternIter
  Perform a heuristically determined sequence of collapse steps (3.3)
  Reduce the set of haplotypes used (3.4)
  If necessary update the best found solution
  repeat
    Execute a precollapse based on the previous best solution (3.5)
    Perform a heuristically determined sequence of collapse steps (3.3)
    Reduce the set of haplotypes used (3.4)
    If necessary update the best found solution
  until no improvements have been obtained in the last
    maxInnerIter (internal) iterations
endfor
If necessary remove the residual variables (3.6)

3.3. The heuristic sequence of collapse steps

As a collapse rule is a generalized version of Clark's rule, Clark's algorithm can be seen as a sequence of collapse rule applications. Its main drawback is due to the specific criterium used to choose the pairs of haplotypes to be collapsed: as one of the two haplotypes must be unambiguous (i.e. must not contain variables) a sufficiently large initial set of unambiguous haplotypes is required to start the algorithm, and even in this case some genotypes may remain unexplained.

We propose a different criterium to choose the haplotypes to be collapsed, which does not require any initial set of unambiguous haplotypes. In contrast to Clark's algorithm, we try to defer the binding of a variable to a specific constant value as much as possible: this is rather intuitive, as the probability of introducing incompatibility increases with the binding of variables to constants, while we are trying to avoid incompatibility in order to perform more collapse steps. Consequently, the pair to be collapsed is chosen from among those which produce the smallest number of variable bindings: a distance matrix is maintained by the algorithm, where the generic cell $M_{i,j}$ represents the number of variables that must be assigned to collapse h_i and h_j. If the haplotypes are incompatible a conventional "infinite" value is assumed.

Given the minimum distance d, one of the pairs at distance d, d+1 and d+2 is randomly chosen. The distance acts as a weight, so that the pairs at distance d have a higher probability of being chosen than those at distance d+2. The sequence of collapse steps stops when the distance matrix contains infinite values for all haplotype pairs.

3.4. Haplotype set reduction

The sequence of collapse rule applications produces a set of haplotypes on which a further reduction can be applied, as some genotypes can be explained in more than one way by using the haplotypes in the set. For each genotype a list of possible solutions is built using the obtained set of haplotypes. The list of each genotype may contain a unique solution (if there is only one pair of extracted haplotypes which can explain it) or some alternatives.

The reduction is based on a greedy technique: a set U of haplotypes is initially built comprising all haplotypes which are in unique solutions. At each step: (i) either at least one genotype with multiple genotype solutions can be explained by a pair of haplotypes in U (and in this case all other solutions are discarded) (ii) or the "best" pair of haplotypes in one of the solution lists is promoted and inserted into U. The weight of a genotype solution is computed taking into account the number of genotype solutions in which the component haplotypes are involved, and the number of other genotype solutions discarded by that solution (a solution usually requires some variable assignments, which produce incompatibilities and hence the discarding of other solutions). Up to 40% of the haplotypes used were removed by this reduction for some problem instances considered in the experiments.

3.5. Precollapsing

When the last iteration has produced an improved solution, a further exploration is initiated partially reusing that solution. In practice, given the previous solution, a subset of the genotype solutions is extracted corresponding to pairs of "good" haplotypes (i.e. haplotypes that are used in several solutions and paired with other good haplotypes). This subset is used to perform a preliminary sequence of collapse steps, followed by a conventional sequence (i.e. driven by the distance matrix).

3.6. Postprocessing: removing residual variables

Even if the algorithm has reached an optimal number of haplotypes some of them may still contain some variables. This corresponds to several (equivalent in terms of the maximum parsimony principle) solutions. In these cases, CollHaps produces a double output: one without variables, where a specific constant value is assigned to each variable, and one with variables (symbolic solution), corresponding to a set of possible solutions. It can be easily shown that:

Proposition 3.2. *Given a symbolic solution, if r is the number of residual variables and k is the number of genotypes whose explanation contains at least one variable, then the number of corresponding distinct solutions is given by 2^{r-k}.*

According to the parsimony principle, each of these 2^{r-k} distinct solutions is optimal and could be randomly assigned. However, in order to obtain better performance in terms of error rates, a different strategy (obviously inspired by the coalescence model) was adopted: given a haplotype containing some variables, the "nearest neighbour" haplotype (i.e. requiring the fewest SNP switches to be transformed into the given haplotype) is searched for and the variables are assigned according to that haplotype.

10.

4. Experimental Results

CollHaps, the program based on the above illustrated algorithm, was tested on a large amount of real biological data and several simulated datasets to study its performance in terms of effectiveness, efficiency and scalability. Its performance was compared with five widely used programs for haplotype inference, namely: (1) Hapinferx, an implementation of Clark's algorithm [Cl90], kindly provided by Prof. Clark. (2) Hapar [WaXu03], a parsimony-optimal program based on a branch and bound algorithm, available at the authors' website. (3) Haplotyper [NQXL02], based on a bayesian algorithm enabling very fast convergence times, available at the authors' website. (4) Phase [StDo03], based on a sophisticated bayesian technique, also available at the authors' website. (5) Gerbil [KiSh05], based on block partitioning and on an expectation-maximization algorithm, available at the authors' website. All programs were run with their default values and Haplotyper was run with the ROUND parameter set to 20, as suggested by the authors. The version 2.1 of the Phase software was used, where relevant improvements were introduced with respect to 1.x versions [StSD01].

4.1. MX1 dataset

Jin et al. [JUD+99] found a 565-bp chromosome 21 region near the MX1 gene, which contains 12 polymorphic sites ($m = 12$). This region is unaffected by recombination and recurrent mutation. Genotypes (not all distinct) from 354 human subjects were arranged into 10 haplotypes forming a perfect phylogeny. The inferred haplotypes were subsequently confirmed by empirical verifications. This set of haplotpyes was also used in [WaXu03] to test the performance of the Hapar algorithm.

We randomly combined the 10 haplotypes to form several datasets of different sizes (number of genotypes) n: 8 sizes ranging from 5 to 19 were considered and 20 distinct datasets were generated for each. The average error rates and average number of haplotypes used are summarized in Table 4.1.

As mentioned above, Hapar is optimal in terms of the number of used haplotypes, as it explores the entire solution space, but CollHaps also found maximum parsimony solutions in 100% of the considered problem instances.

For low n values, most haplotypes are used only once in the solution and all algorithms based solely on the maximum parsimony principle have higher error rates. In contrast, Phase, which incorporates the coalescent model, and Gerbil, which assumes rare recombination events, outperform all other algorithms, but the percentage of incorrectly inferred haplotpyes is nevertheless very relevant. This is obviously motivated by the fact that most algorithms use a significantly lower number of haplotypes to explain the genotypes and the correct solution is actually one of the worst in terms of parsimony.

For values of n near or larger than m, i.e. when each haplotype is used at least twice (on average) in the solution, parsimony leads to good solutions and error rates tend to zero. For n values above 13, Gerbil, Hapar and CollHaps have the best performance and CollHaps is the only algorithm to achieve an average (both genotype and switch) error rate below 1% for $n = 19$.

All algorithms were able to solve each problem instance in less than one second, except Phase which required processing times ranging from 3 to 18 seconds.

n	5	7	9	11	13	15	17	19
				average genotype error rate				
Gerbil	**.530**	.350	**.200**	**.141**	**.119**	.090	.044	.011
Hapinferx	.760	.636	.539	.514	.423	.430	.415	.400
Hapar	.660	.521	.344	.241	.135	**.060**	**.038**	.018
Haplotyper	.640	.543	.439	.300	.165	.087	.053	.029
Phase	.610	**.329**	.228	.177	**.119**	.073	.065	.018
CollHaps	.630	.514	.328	.186	.127	.067	**.038**	**.008**
				average switch error rate				
Gerbil	**.667**	.435	**.234**	**.153**	**.118**	.089	**.046**	.012
Hapinferx	.898	.728	.617	.597	.502	.515	.504	.491
Hapar	.846	.661	.457	.286	.156	**.068**	.047	.019
Haplotyper	.823	.696	.550	.340	.190	.102	.065	.036
Phase	.701	**.380**	.257	.185	.145	.081	.086	.021
CollHaps	.786	.612	.409	.239	.132	.079	.048	**.008**
				average number of haplotypes used				
Gerbil	7.70	9.60	10.00	10.35	10.45	10.55	10.30	10.15
Hapinferx	7.95	9.85	11.40	12.70	12.65	13.10	14.15	15.00
Hapar	**6.95**	**8.45**	**9.25**	**9.70**	**9.80**	**10.00**	**10.00**	**10.00**
Haplotyper	**6.95**	**8.45**	9.40	9.80	9.90	10.05	10.05	10.05
Phase	7.65	9.50	9.70	10.65	10.50	10.35	10.45	10.20
CollHaps	**6.95**	**8.45**	**9.25**	**9.70**	**9.80**	**10.00**	**10.00**	**10.00**

The results for the best-performing algorithms in each column are in bold.

Table 1: Performance comparison on MX1 data ($m = 12$)

4.2. Cystic Fibrosis Transmembrane-Conductance Regulator (CFTR) Gene dataset

Cystic fibrosis is one of the most common autosomal recessive diseases in Caucasian populations, occurring approximately once in 2000 live births. Nearly 70% of mutations in cystic fibrosis patients were shown to correspond to a specific 3-bp deletion in the CFTR gene on chromosome 7 (region q31) by Kerem et al. [KRB+89], who collected data from affected and healthy individuals on 23 SNPs in a 1.8 Mb candidate regions.

As in [NQXL02] we considered a subset of 57 haplotypes (29 of which are distinct) with no missing data from the 94 experimentally identified disease haplotypes. The haplotypes were randomly combined to form several datasets of different sizes. In [NQXL02, StDo03] only datasets of size 28 were used to test the performance of Haplotyper and Phase. The high error rates obtained in the experiments were attributed to the low number of genotypes with respect to the number of distinct haplotypes. In our experiments 8 sizes (ranging from 28 to 56 genotypes) were therefore considered and for each size, 20 distinct datasets were generated.

The average switch error rates and number of haplotypes used are illustrated in Table 4.2 (the relative performance of the algorithms in terms of switch and genotype error rate are very similar). Hapar was run on some sample instances without ever finding a solution (the program was stopped after one day of processing).

As with the previous example, CollHaps has the best performance in terms of number of haplotypes used, while Phase has the best error rates for low n values. For large n values, the error rates of all programs progressively decrease and CollHaps outperforms the other algorithms for both the genotype and the switch error rate.

n	28	32	36	40	44	48	52	56
				average genotype error rate				
Gerbil	.579	.550	.531	.504	.472	.450	.454	.442
Hapinferx	.941	.892	.836	.795	.743	.703	.681	.677
Hapar	-	-	-	-	-	-	-	-
Haplotyper	.298	.275	.221	.138	.118	.098	.082	.079
Phase	**.264**	**.195**	**.153**	**.114**	**.091**	.086	.079	.069
CollHaps	.355	.289	.193	.130	.109	**.085**	**.069**	**.059**
				average switch error rate				
Gerbil	.345	.316	.307	.300	.280	.264	.265	.255
Hapinferx	.930	.840	.770	.725	.660	.618	.593	.586
Hapar	-	-	-	-	-	-	-	-
Haplotyper	.185	.160	.134	.085	.071	.058	.049	.047
Phase	**.154**	**.114**	**.090**	**.066**	**.054**	**.048**	.045	.039
CollHaps	.207	.184	.125	.087	.073	.051	**.041**	**.037**
				average number of haplotypes used				
Gerbil	39.00	40.95	44.50	47.35	49.30	50.35	52.05	53.20
Hapinferx	52.75	55.35	58.25	62.55	63.75	66.10	68.05	72.40
Hapar	-	-	-	-	-	-	-	-
Haplotyper	26.50	27.45	27.85	28.25	28.55	28.55	28.70	28.85
Phase	28.55	28.60	28.75	28.85	28.90	29.05	29.10	29.16
CollHaps	**26.40**	**27.15**	**27.70**	**28.15**	**28.50**	**28.50**	**28.55**	**28.75**

For some sample instances, Hapar found no solutions within one day.
The results for the best-performing algorithms in each column are in bold.

Table 2: Performance comparison on CFTR data ($m = 23$)

Gerbil, Haplotyper and Collhaps have similar processing times (all from 3 to 10 seconds), while Phase requires some minutes (from 4 to 10) to process a single instance. Hapinferx is very fast (instances are always processed in less than one second) but has unacceptable error rates. Gerbil error rates are also particularly high; this may be related to the characteristics of these haplotypes and Gerbil's assumption of rare recombination events in blocks.

4.3. Datasets Generated Using Hudson's software

The performance of CollHaps was also tested on some simulated data. The parent haplotypes were generated using Hudson's software ms [Hu02], in particular a different collection of 30 haplotypes was generated for each dataset. As in [BrHa04] the duplicate haplotypes were eliminated to produce harder input sets, where completely resolved genotypes are less likely. We considered datasets with recombination level r values of 0, 4, 16 and 40, as in [Gu03], and 100, as in [WaXu03]. For each value 20 datasets were generated, each with 30 genotypes and 10 sites. The results are summarized in Table 4.3.

Obviously Hapar always produced maximum parsimony solutions, but this was also achieved by CollHaps in all considered problem instances. However, Hapar required more than 2 hours of processing time to solve two of the instances with $r = 100$, while Collhaps solved all instances in less than one second. Processing times under one second were also achieved by Gerbil, Hapinferx and Haplotyper, while Phase required from 20 to 80 seconds. CollHaps outperforms all other algorithms in terms of both genotype and switch error rate.

Recombination levels r	0	4	16	40	100
			average switch error rate		
Gerbil	**.000**	.009	.124	.167	.222
Hapinferx	.223	.284	.414	.399	.511
Hapar	**.000**	.008	.048	.084	.123
Haplotyper	**.000**	.007	.058	.125	.161
Phase	**.000**	.014	.049	.107	.142
CollHaps	**.000**	**.004**	**.045**	**.072**	**.092**
			average number of haplotypes used		
Gerbil	**7.25**	9.20	12.65	15.95	19.60
Hapinferx	11.15	14.45	17.65	19.70	24.95
Hapar	**7.25**	**8.90**	**10.80**	**13.40**	**15.40**
Haplotyper	**7.25**	8.95	10.85	13.75	15.80
Phase	**7.25**	8.95	11.05	14.25	16.70
CollHaps	**7.25**	**8.90**	**10.80**	**13.40**	**15.40**

The results for the best-performing algorithms in each column are in bold.

Table 3: Performance comparison on simulated data ($n = 30$, $m = 10$)

4.4. Angiotensin Converting Enzyme Dataset

Angiotensin Converting Enzyme (encoded by the gene DCP1, also known as ACE) catalyzes the conversion of angiotensin I to the physiologically active peptide angiotensin II, which controls fluid-electrolyte balance and systemic blood pressure. Rieder et al. [RTCN99] completed the genomic sequencing of DCP1 from 11 individuals, identifying 78 varying sites in 22 chromosomes that were resolved into 13 distinct haplotypes. As in [WaXu03, StDo03] we only considered the 52 biallelic sites of these haplotypes.

	Number of used haplotypes	Average genotype error rate	Average switch error rate	Average proc. time (in sec.)
Gerbil	**13**	.181	**.042**	7.8
Hapinferx	**13**	.272	.073	**0.1**
Hapar	**13**	.272	.063	78.8
Haplotyper	**13**	**.159**	.057	6.1
Phase	**13**	.181	.063	116.2
CollHaps	**13**	.272	.053	0.4

The results for the best-performing algorithms in each column are in bold. CollHaps produced a partially symbolic solution corresponding to 2^{17} distinct parsimony-equivalent solutions.

Table 4: Performance comparison on ACE dataset ($n = 11$, $m = 52$)

We performed 20 runs for each program (each time changing the random seed when possible). The results are summarized in Table 4.4. All algorithms correctly found maximum parsimony

solutions using 13 haplotypes, but the genotype error rate is nevertheless quite high for all algorithms. The symbolic solution produced by CollHaps accounts for such poor performance: 3 out of the 11 haplotype pairs are partially symbolic (contain variables), corresponding to 2^{17} (see Proposition 3.2 above) distinct solutions, all using 13 haplotypes and hence parsimony-equivalent. Furthermore, these 3 haplotype pairs have no variable in common with any other haplotype, thus requiring the introduction of 6 distinct and unique haplotypes.

4.5. Chromosome 5q31 dataset

Daly et al. [DRS+01] studied a 500-kb region on human chromosome 5q31, which is implicated as containing a genetic risk factor for Crohn's disease. The original diploid data contain 129 pedigrees (mother, father and child) each genotyped at 103 SNPs. As in [KiSh05] we considered the 258 haplotypes from the children and used them to generate datasets of size (number of genotypes) 500, 600, 700, 800, 900 and 1000. To produce more difficult input sets, where completely resolved genotypes are less likely, the duplicate haplotypes were eliminated and only 178 unique haplotypes were considered for random sampling and pairing.

n	500	600	700	800	900	1000
			switch error rate			
Gerbil	.26853	.26724	.27347	.28444	.28396	.28317
Hapinferx	.16587	.16056	.16199	.16242	.16061	.15734
Phase	.00116	.00035	.00030	.00013	**.00000**	.00011
CollHaps(30)	**.00032**	**.00018**	**.00008**	**.00007**	.00006	**.00005**
CollHaps(3)	**.00032**	**.00018**	**.00008**	**.00007**	.00006	**.00005**
			number of haplotypes used			
Gerbil	745	851	981	1100	1191	1320
Hapinferx	418	466	518	570	632	668
Phase	180	180	180	**178**	**178**	**178**
CollHaps(30)	**178**	**178**	**178**	**178**	**178**	**178**
CollHaps(3)	**178**	**178**	**178**	**178**	**178**	**178**

Hapar and Haplotyper were unable to obtain any solutions.
The results for the best-performing algorithms in each column are in bold.

Table 5: Performance comparison on 5q31 dataset ($m = 103$)

Hapar and Haplotyper were unable to obtain any solutions. CollHaps performance was measured using its default parameter values (processing times ranging from 2 to 4 hours) and with the number of iterations set to 3 (the default is 30) to achieve processing times below 30 minutes for all instances. Gerbil required processing times of 15 to 25 minutes, while Hapinferx required 5 minutes at most to process each problem instance. Phase processing times ranged from 12 hours to one day.

Phase and CollHaps performance is clearly superior to the other programs in terms of both parsimony and error rates. Collhaps was always able to obtain a solution with 178 haplotypes, even with only 3 iterations, and outperformed Phase error rates in 5 of the 6 instances.

5. Conclusion

In this paper we presented a new algorithm for the haplotype inference problem, based on the maximum parsimony principle and a generalized version of Clark's rule, named collapse rule. Some relevant properties of this rule were illustrated, particularly that any optimal solution can be obtained by a suitable sequence of collapse rule applications. Finally, the implementation of the proposed algorithm (CollHaps) was tested on several real biological and simulated datasets. The experiments clearly show that CollHaps achieves good performance in terms of parsimony (number of haplotypes used), error rates and processing times. We are currently working on further developments of the software to cope with incomplete data and identify blocks in haplotype sequences. We are also planning to use the algorithm to determine good upper bounds in parsimony-optimal techniques, based on Integer Linear Programming formulations of the problem.

References

[BrHa04] Brown,D.G. and Harrower,I.M. (2004) A New Integer Programming Formulation for the Pure Parsimony Problem in Haplotype Analysis, In *Proc. of the 4th Workshop on Algorithms in Bioinformatics (WABI 2004)*, 254-265.

[Cl90] Clark,S. (1990) Inference of Haplotypes from PCR-amplified Samples of Diploid Populations, *Mol.Biol.Evol*, **7**(2), 111-122.

[CoBC98] Collins,F.S., Brooks,L.D. and Chakravarti,A. (1998) A DNA Polymorphism Discovery Resource for Research on Human Genetic Variation, *Genome Research*, **8**(12), 1229-1231.

[DRS+01] 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, *Nature Genetics*, **29**, 229-232.

[Gu03] Gusfield,D. (2003) Haplotype Inference by Pure Parsimony, In *Proc. of the 14th Annual Symposium on Combinatorial Pattern Matching (CPM 2003)*, 144-155.

[Gu04] Gusfield,D. (2004) An Overview of Combinatorial Methods for Haplotype Inference, In *Computational Methods for SNPs and Haplotype Inference*, LNCS 2983, 9-25.

[HBE+04] Halldorsson,B.V., Bafna,V., Edwards,N., Lippert,R., Yooseph,S. and Istrail,S. (2004) A Survey of Computational Methods for Determining Haplotypes, In *Computational Methods for SNPs and Haplotype Inference*, LNCS 2983, 26-47.

[HaKi95] Hawley,M.E. and Kidd,K.K. (1995) HAPLO: a program using the EM algorithm to estimate the frequencies of multi-site haplotypes, *J Hered.*, **86**, 409-411.

[HKW+00] Hoehe,M.R., Köpke,K., Wendel,B., Rohde,K., Flachmeier,C., Kidd,K.K., Berrettini,W.H. and Church G.M. (2000) Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence, *Human Molecular Genetics*, **9**(19), 2895-2908.

[Hu02] Hudson,R.R. (2003) Generating samples under a Wright-Fisher neutral model of genetic variation, *Bioinformatics*, **18**(2), 337-338.

[JUD+99] Jin,L., Underhill,P.A., Doctor,V., Davis,R.W., Shen,P. Cavalli-Sforza,L.L. and Oefner,P.J. (1999) Distribution of haplotypes from a chromosome 21 region distinguishes multiple prehistoric human migiration, *Proc. Natl. Acad. Sci. USA*, **96**, 3796-3800.

[KRB+89] Kerem,B., Rommens,J.M., Buchanan,J.A., Marliewicz,D., Cox,T.K., Chakravarti,A., Buchwald,M. and Tsui,L. (1989) Identification of the Cystic Fibrosis Gene: Genetic Analysis, *Science*, **245**, 1073-1080.

[KiSh05] Kimmel,G. and Shamir,R. (2005) GERBIL: Genotype resolution and block identification using likelihood, *Proc. Natl. Acad. Sci. USA*, **102**(1), 158-162.

[LaPR04] Lancia,G., Pinotti,M.C. and Rizzi, R. (2004) Haplotyping Populations by Pure Parsimony: Complexity of Exact and Approximation Algorithms, *INFORMS Journal on Computing*, **16**(4), 348-359.

[LCZC02] Lin,S., Cutler,D.J., Zwick,E. and Chakravarti,A. (2002) Haplotype Inference in Random Population Samples, *American Journal of Human Genetics*, **71**, 1129-1137.

[LWU95] Long,J.C., Williams,R.C. and Urbanek,M. (1995) An E-M algorithm and testing strategy for multiple-locus haplotypes, *American Journal of Human Genetics*, **56**, 799-810.

[NQXL02] Niu,T., Qin,Z.S., Xu,X. and Liu,J.S. (2002) Bayesian Haplotype Inference for Multiple Linked Single-Nucleotide Polymorphisms, *American Journal of Human Genetics*, **70**, 157-169.

[RTCN99] Rieder,M.J., Taylor,S.L., Clark,A.G. and Nickerson,D.A. (1999) Sequence variation in the human angiotensin converting enzyme, *Nature Genetics*, **22**, 59-62.

[StDo03] Stephens,M. and Donnelly,P. (2003) A comparison of bayesian methods for haplotype reconstruction from population genotype data, *American Journal of Human Genetics*, **73**(5), 1162-1169.

[StSD01] Stephens,M., Smith,N. and Donnelly,P. (2001) A new statistical method for haplotype reconstruction from population data, *American Journal of Human Genetics*, **68**, 978-989.

[Ven+01] Venter,J.C. et al (2001) The Sequence of the Human Genome, *Science*, **291** 1304-1351.

[WaXu03] Wang,L. and Xu,Y. (2003) Haplotype Inference by Maximum Parsimony, *Bioinformatics*, **19**(14), 1773-1780.