A Class Representative Model for Pure Parsimony Haplotyping

Daniele Catanzaro[†], Alessandra Godi[*] and Martine Labbé[†]

June 5, 2008

Abstract

Haplotyping estimation from aligned Single Nucleotide Polymorphism (SNP) fragments has attracted more and more attention in the recent years due to its importance in analysis of many fine-scale genetic data. Its application fields range from mapping of complex disease genes to inferring population histories, passing through designing drugs, functional genomics and pharmacogenetics. The literature proposes several criteria for haplotyping populations, each of them characterized by biological motivations. One of the most important haplotyping criteria is the parsimony which consists of finding the minimum number of haplotypes necessary to explain a given set of genotypes. Parsimonious haplotype estimation is a $\mathcal{NP}$-Hard problem for which the literature has proposed several Integer Programming (IP) models. Here we describe a new polynomial-sized IP model based on the concept of class representatives, already used for the coloring problem. We propose valid inequalities to strengthen our model and show, through computational experiments, that our model outperforms the best IP models currently known in literature.

Keywords: haplotype inference, computational biology, integer programming.

1 Introduction

Diploid organisms, such as humans, are characterized by having the DNA organized in pairs of chromosomes, thereof one copy is inherited from the father and the other from the mother. The recent completion of the sequencing phase of the Human Genome Project (Venter (2001)) showed that such copies are extremely similar among them, and that the genomes of two different individuals are identical in more than 99% of the overall number of nucleotides. Nevertheless, differences at genomic level (also known as *polymorphisms*) occur, on average, every thousand bases (Chakravarti (1998)), and are (excluding the recombination process) the predominant form of human variation as well as of genetic diseases (Hoehe et al. (2000); Terwilliger and Weiss (1998)).

When a site (i.e., the position of a specific nucleotide) of the genome shows a statistically significant variability within a population (i.e., a set of individuals) then it is called *Single Nucleotide Polymorphism* (SNP). Specifically, a site is considered a SNP if for a minority (at least 5%) of the population a certain nucleotide is observed (called the least frequent allele) while for the rest of the population a different nucleotide is observed (the most frequent allele). For a given SNP, an individual can be either homozygous (i.e., possess the same allele on both chromosomes) or heterozygous (i.e., possess two different alleles). The values of a set of SNPs on a particular chromosome copy define a *haplotype*. Haplotyping an individual therefore consists of determining, for each copy of a given chromosome, a pair of haplotypes.

Haplotyping populations of individuals has attracted more and more attention in the recent years (Helmuth (2001); Marshall (1999)) due to its importance in analysis of many fine-scale genetic data (Clark et al. (1998); Schwartz et al. (2002)). For example, haplotypes are necessary in evolutionary studies to extract the information needed to detect diseases and to reduce the number of tests to be carried out. In functional genomics haplotypes are used to discover a functional gene or in the study of an altered response of an organism to a particular therapy. In human pharmacogenetics, haplotypes explain why people react differently

[*] Istituto di Analisi dei Sistemi e Informatica, Consiglio Nazionale delle Ricerche (C.N.R.), viale Manzoni 30, Roma, Italy. Phone: 0039 6 77161. Fax: 0039 6 7716431.

[†] Graphs and Mathematical Optimization (G.O.M.), Computer Science Department, Université Libre de Bruxelles (U.L.B.), Boulevard du Triomphe, CP 210/01, B-1050, Brussels, Belgium. Phone: 0032 2 650 5628. Fax: 0032 2 650 5970.

Correspondence should be addressed to: dacatanz@ulb.ac.be

to different types or amounts of drugs. In fact, since SNPs affect the structure and function of proteins and enzymes, they may influence the way in which a drug is absorbed and metabolized.

Direct sequencing of haplotypes via experimental methods is both time-consuming and expensive therefore current molecular sequencing methods generally provide a cheaper genotype information. Specifically, genotype data provide information about the multiplicity of each SNP allele of a given individual, i.e., the knowledge about its homo or heterozygous nature. Unfortunately, a drawback of using genotype data is that information regarding which heterozygous site SNP variants came from the same chromosome copy remains unknown (Wang and Xu (2003)). Hence, in silico haplotyping methods become attractive alternatives, and in some cases the only viable way for haplotyping populations (Lancia et al. (2004)).

The simplest way for haplotyping a population is described in (Bonizzoni et al. (2003); Gusfield (2003); Lancia et al. (2004)) and can be resumed as follows: first, to obtain experimentally genotype data, and subsequently, for each individual, to retrieve the haplotypes computationally, i.e., to find a set of haplotypes such that, if they are assumed to be the corresponding set of chromosome copies, then, computing the multiplicity of each SNP allele one can obtain exactly the genotypes given. However, this approach requires the presence of some haplotyping criterion.

Several criteria have been proposed for haplotyping populations, each of them based on biological motivations (see for example Bafna et al. (2003); Clark et al. (1998); Eskin et al. (2003); Excoffier and Slatkin (1995); Fallin and Schork (2000); Lancia and Rizzi (2006); Niu et al. (2002a,b); Stephens and Donnelly (2003); Stephens et al. (2001)). In this article we consider the *parsimony* criterion (Lancia et al. (2004)). The idea at the core of the parsimony criterion is that under many plausible explanations of an observed phenomenon, the one requiring the fewest assumptions should be preferred (Semple and Steel (2003)). Hence, since the number of distinct haplotypes observed in a population is much smaller than the number of possible haplotypes, the parsimony based approaches aims to determine the minimum number of different haplotypes that, combined in pairs over time, have given rise to a set of observed genotypes (Lancia et al. (2004)).

The problem of haplotyping populations under parsimony (hereafter denoted as the Pure Parsimony Haplotyping (PPH) problem) is known to be $\mathcal{NP}$-hard and $\mathcal{APX}$-hard (Lancia et al. (2004)). This result led the development of several enumerative optimization algorithms aiming to solve exactly instances of PPH. Specifically, Gusfield (2003) first proposed an integer-programming model to tackle instances of PPH. The author described a polynomial-size model characterized by two kinds of variables, one for haplotypes and the other for haplotype-pairs, and by the exhaustive generation of the set of all haplotypes compatible with some genotype in the input. Similar integer programming models were also employed by Brown and Harrower (2004) and Bertolazzi et al. (2008). In order to minimize the number of distinct haplotypes, (Brown and Harrower (2004)) proposed to construct haplotype vectors by associating a variable to each site; subsequently they used constraints to establish the exact haplotype structures. Differently, (Bertolazzi et al. (2008)) first formulated PPH as a minimization problem characterized by a polynomial number of variables and constraints; then, the authors turned the problem into a maximization problem and strengthened the model by using clique inequalities, symmetry-breaking inequalities and dominance relations. While Gusfield (2003) and Bertolazzi et al. (2008) used commercial MIP solvers (Cplex and Xpress-MP, respectively) to get solutions of their models, Brown and Harrower (2004) used a branch-and-cut algorithm to solve their polynomial model. A comparison of their results shows that Brown and Harrower (2004) polynomial model is well suited for big dimension samples, whereas Gusfield (2003) and Bertolazzi et al. (2008) ones are more efficient for medium dimension, and specifically, when the recombination level (i.e., the parameter that affects the structure of the haplotypes) increases. So far, to the authors best knowledge, datasets containing 68 genotypes and 75 SNPs represent the limit size instances of PPH that can be exactly analyzed (Brown and Harrower (2006)).

In this article we present a new integer programming (IP) model for PPH. Interpreting PPH as a specific covering problem, we provide a new polynomial-sized IP model by using the idea of class representatives already proposed by Campelo et al. (2008) for the coloring problem (Garey and Johnson (2003)). Computational experiments show that our model outperforms the existing polynomial IP models for PPH (Bonizzoni et al. (2003); Brown and Harrower (2006); Gusfield (2003); Lancia et al. (2004)) both on real and simulated data, and allows haplotyping of populations containing hundreds of genotypes and SNPs. The model is compact, easy to implement, solvable with standard solvers, and usable in those cases for which the parsimony criterion is well suited for haplotyping populations.

Chromosome paternal:	ataggtccCtatttccaggcgcCgtatacttcgacgggActata
Chromosome maternal:	ataggtccGtatttccaggcgcCgtatacttcgacgggTctata
Haplotype 1 →	C C A
Haplotype 2 →	G C T

Table 1: An example of haplotypes and SNP.

2 Notation and Problem Formulation

Assume a population of diploid organisms is given, and that a pair (paternal and maternal) of chromosomes of a specific organism is available. Then, all sites in this pair of chromosomes at which some organisms have one nucleotide whereas others have a different one, are called single nucleotide polymorphisms. If both nucleotides are equal, a SNP is said homozygous, otherwise it is said heterozygous. An example from (Lancia et al. (2004)), Table 1 shows three SNPs, thereof two heterozygous (the first and the third SNPs) and one homozygous (the second SNP).

The set of SNPs of a paternal (maternal) chromosome defines a haplotype. Due to the binary nature (homo or heterozygous) of a SNP, haplotypes can be seen as strings of equal size over an alphabet $\Sigma = \{0, 1\}$. For example, assume we encode the first SNPs in Table 1 (i.e., 'C' and 'G') with 0 and 1, respectively; the second SNPs (i.e., 'C' and 'C') with 1; and the third SNPs (i.e., 'A' and 'T') with 0 and 1, respectively. Then, we may encode haplotype 1 as $(0, 1, 0)$ and haplotype 2 as $(1, 1, 1)$.

Since at each SNP only three possibilities can arise (either homozygous of type 0/1 or heterozygous) then a genotype can be represented as strings over an alphabet $\Sigma = \{0, 1, 2\}$, where entries equal to 0 (1) indicate homozygous sites of type 0 (1), and entries equal to 2 indicate heterozygous sites.

Given a pair of haplotypes $\{h_i, h_j\}$, define the operator sum $\oplus$ among h_i and h_j as the genotype g whose p-th entry is h_{ip} if $h_{ip} = h_{jp}$, and 2 otherwise. As an example, the genotype obtained by summing haplotype 1 and 2 in Table 1 is $g = (212)$. We say that a genotype g_k is resolved from a pair of haplotypes $\{h_i, h_j\}$ if $g_k = h_i \oplus h_j$.

An instance and the corresponding solution of PPH are specified by indicating a set $\mathcal{G}$ of m genotypes and a minimal set $\mathcal{H}$ of at most $2m$ haplotypes Lancia et al. (2004) such that for each genotype $g_k \in \mathcal{G}$ there exists a pair of haplotypes $\{h_i, h_j\}$ resolving g_k. Hereafter, we denote the set of genotypes $g_k \in \mathcal{G}$ as K.

3 Integer Programming Models

In this Section we describe a possible Integer Programming (IP) model for PPH. First, we describe a basic model which requires a polynomial number of variables and constraints. Subsequently we show how to reduce the model size by exploiting the particular structure and properties of variables involved. Finally, we provide valid inequalities to further strengthen the model.

3.1 Basic Model

Denote $\mathcal{P}$ as the set of the p SNPs characterizing each genotype g_k in $\mathcal{G}$ and consider a feasible solution of PPH: it consists of a set $\mathcal{H}$ of haplotypes such that each one is used to resolve a subset of genotypes. Then, this solution induces a covering of the genotypes by subsets such that: (i) each subset of genotypes shares one haplotype, (ii) each genotype belongs to exactly two subsets, and (iii) every pair of subsets intersects in at most one genotype. As an example, the five genotypes in Figure 1 can be covered by four subsets induced by four haplotypes $h_1 = \{1011\}$, $h_{1'} = \{1101\}$, $h_2 = \{0011\}$, $h_{2'} = \{1000\}$.

This observation led us to develop an integer programming model for PPH based on the class representatives with smallest index, similar to the one proposed by Campelo et al. (2008) to tackle the coloring problem (Garey and Johnson (2003)). Following this rationale, we say that each haplotype $h \in \mathcal{H}$ induces a subset of genotypes S such that each genotype g_k belongs to exactly two such subsets (since it must be explained by exactly two haplotypes).

We also associate an index to each subset S of genotypes induced by a haplotype h. Specifically, if i is the smallest index of a genotype belonging to S, then i is the index associated to S and the subset will

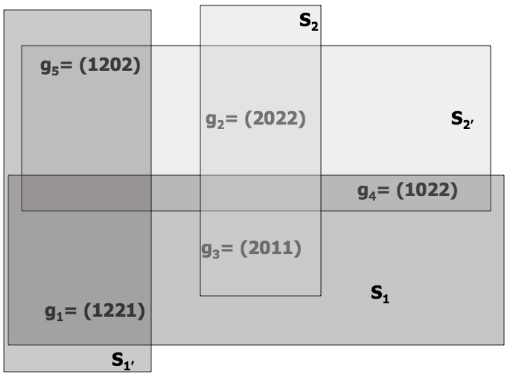

Figure 1: An example of a covering of five genotypes induced by a set of four haplotypes.

be denoted as S_i. Since each genotype k belongs to exactly two subsets it may happen that k is itself the genotype with smallest index in both subsets. In this case a dummy genotype k' is added, and the subset $S_{k'}$ is created. As an example, one can imagine that the haplotype h_1 induces the subset $S_i = \{g_i, g_j, g_k, \ldots\}$, h_2 induces the subset $S_{i'} = \{g_i, g_l, g_r, g_s, \ldots\}$, h_3 induces the subset $S_k = \{g_k, g_l, g_s, g_t, \ldots\}$, and so on. We remark that the index k' can be considered only if k was previously used, i.e., if the subset S_k already exists.

Since at most $2m$ haplotypes are necessary to resolve m genotypes (Lancia et al. (2004)), then the indices i of the subsets S_i can vary inside $K \cup K'$, where $K = \{1, \ldots, m\}$ and $K' = \{1', \ldots, m'\}$. Assume that an order is defined on $K \cup K'$ in such a way that $1 < 1' < 2 < 2' < \ldots < m < m'$. Define x_i, $\forall\, i \in K \cup K'$, as a decision variable equal to 1 if, in the solution, there exists a haplotype inducing a subset S_i of genotypes whose smallest index genotype is g_i, and 0 otherwise. Denote y_{ij}^k, $\forall\, k \in K, \forall\, i, j \in K \cup K'$, $i < j$, as a decision variable equal to 1 if the genotype k belongs to the subsets S_i and S_j, and 0 otherwise. Finally denote z_{ip}, $\forall\, i \in K \cup K'$, $p \in \mathcal{P}$, as a decision variable equal to 1 if the haplotype inducing the subset S_i of genotypes has such a value at p-th site, and 0 otherwise. Variables z_{ip} shall describe explicitly the haplotypes of the solution. Then, a possible integer programming model for PPH can be stated as follows:

Formulation 1. - *Basic Model (BM)*

$$\min \sum_{i \in K \cup K'} x_i \tag{1}$$

$$s.t.\ x_{i'} \le x_i \qquad \forall\, i \in K \tag{2}$$

$$\sum_{i,j \in K \cup K'} y_{ij}^k \ge 1 \qquad \forall\, k \in K \tag{3}$$

$$\sum_{j \in K \cup K' : j \ge i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \le x_i \qquad \forall\, k \in K, \forall\, i \in K \cup K' \tag{4}$$

$$y_{kk'}^k \le x_{k'} \qquad \forall\, k \in K \tag{5}$$

$$z_{kp} = z_{k'p} = 0 \qquad \forall\, k \in K, \forall\, p \in \mathcal{P} : g_{kp} = 0 \tag{6}$$

$$z_{kp} = z_{k'p} = 1 \qquad \forall\, k \in K, \forall\, p \in \mathcal{P} : g_{kp} = 1 \tag{7}$$

$$z_{kp} + z_{k'p} = 1 \qquad \forall\, k \in K, \forall\, p \in \mathcal{P} : g_{kp} = 2 \tag{8}$$

$$z_{ip} \le 1 - \sum_{j \in K \cup K' : j \ge i} y_{ij}^k - \sum_{j \in K \cup K' : j < i} y_{ji}^k \qquad \forall\, p \in \mathcal{P}, \forall\, k \in K : g_{kp} = 0, \forall\, i \in K \cup K', i \ne k, k' \tag{9}$$

$$z_{ip} \ge \sum_{j \in K \cup K' : j \ge i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \qquad \forall\, p \in \mathcal{P}, \forall\, k \in K : g_{kp} = 1, \forall\, i \in K \cup K', i \ne k, k' \tag{10}$$

$$z_{ip} + z_{jp} \ge y_{ij}^k \qquad \forall\, p \in \mathcal{P}, \forall\, k \in K : g_{kp} = 2, \forall\, i, j \in K \cup K' \tag{11}$$

$$z_{ip} + z_{jp} \le 2 - y_{ij}^k \qquad \forall\, p \in \mathcal{P}, \forall\, k \in K : g_{kp} = 2, \forall\, i, j \in K \cup K' \tag{12}$$

$$x_i,\ z_{ip},\ y_{ij}^k \in \{0, 1\}. \tag{13}$$

The objective function (1) represents the number of distinct haplotypes or equivalently the cardinality of $\mathcal{H}$. Since the index i' is considered only if i is already used, constraints (2) implies that if the haplotype

h_i is not used, then $h_{i'}$ should not be used. Constraints (3) impose that each genotype g_k must belong to exactly two subsets S_i, S_j, and constraints (4) force x_i to be 1, i.e., to take haplotype h_i into account, if some genotype g_k is resolved by h_i. Constraints (5) are a consequence of the definition of the dummy genotype k'. Actually, they constitute a special version of constraints (4) when genotype k is resolved by haplotype k'. Constraints (6-8) translate the sum operation among haplotypes. Specifically: constraints (6) impose that p-th SNP of the haplotype h_i ($h_{i'}$), inducing the subset S_i ($S_{i'}$), must be set to 0 when genotype g_i has its p-th SNP is equal to 0. By analogy, constraints (7) impose that p-th SNP of the haplotype h_i ($h_{i'}$) must be set to 1 when genotype g_i has its p-th SNP is equal to 1. Constraints (8) impose that exactly one among the p-th SNPs of the haplotypes h_i and $h_{i'}$ can be set to 1 when genotype g_i has its p-th SNP is equal to 2. Constraints (9) establish the relations between variables z_{ip} and y_{ij}^k. Specifically, they force the p-th SNP of the haplotype h_i to be equal to 0 when at least one genotype g_k, whose p-th entry equal to 0, belongs to the induced subset S_i. By analogy, constraints (10) force the p-th SNP of the haplotype h_i to be equal to 1 when at least one genotype g_k, whose p-th entry equal to 1, belongs to the induced subset S_i. Finally, constraints (11-12) impose that the sum operation among haplotypes is respected for any pair of haplotypes h_i and h_j in the solution.

3.2 Reducing Model Size

It is worth noting that the particular nature of the set of indices $K \cup K'$ can be exploited to reduce the size of BM. In fact, given that $y_{ij}^k = 1$ if and only if k belongs to two subsets having g_i and g_j for smallest index genotype, we need to define variables y_{ij}^k only when $i < j \leq k$ or $i = k$ and $j = k'$. For example, variable $y_{1,1'}^2$ does not need to be defined as well as all variables $y_{ii'}^k$ for all $i, k \in K$, $k \neq i$. Similarly, variables $y_{ik'}^k$ or $y_{k'i}^k$ (depending on whether k is smaller or bigger than i) do not need to be defined for $i \in K \cup K'$ with $i \neq k$ and $i \neq k'$. In fact, if $y_{ik'}^k = 1$, then k belongs to two subsets, one represented by i and the other one by k', which contradicts the assumption that the dummy genotype k' can be considered only if k is already used. By extending this analysis to all the possible cases in which variables y_{ij}^k are redundant and assuming that variable $y_{11'}^1 = 1$, it is easy to see that variables y_{ij}^k do not need to be defined whenever they belongs to one of the following sets:

$$\begin{aligned} R_1 &= \{y_{ij}^k : k \in K,\ i,j \in K \cup K', j < i < k\} && (14)\\ R_2 &= \{y_{ik'}^k : k \in K,\ i \in K \cup K', i \leq (k-1)'\} && (15)\\ R_3 &= \{y_{ii'}^k : k \in K,\ i \in K \cup K', 2 \leq i \leq k-1\}. && (16)\end{aligned}$$

The sets of redundant variables can be further expanded by observing that for each triplet of genotypes $\{g_i, g_j, g_k\}$ such that the respective p-th SNP is $g_{ip} = 0$, $g_{jp} = 0$, and $g_{kp} = 2$, variable y_{ij}^k is necessarily equal to 0 since the belongings of the genotype g_k to the subsets S_i and S_j would violate the sum operator among haplotypes at least at p-th SNP. Extending this argument to all the possible combinations of triplets of genotypes violating the haplotype sum operator, it is easy to see that the following proposition, whose proof is omitted, holds:

Proposition 1. *The set of variables*

$$R_4 = \{y_{ij}^k :\ i,j,k \in K, p \in \mathcal{P}, g_{kp} = 2, g_{ip} = g_{jp} \neq 2\} \qquad (17)$$

is redundant.

Similar arguments can be used to prove the following proposition:

Proposition 2. *The set of variables*

$$R_5 = \{y_{ij}^k :\ i,j,k \in K, p \in \mathcal{P}, g_{kp} + g_{ip} = 1 \textit{ or } g_{kp} + g_{jp} = 1\} \qquad (18)$$

is redundant.

Proof. By contradiction, if the set R_5 is not redundant then, for some $p \in \mathcal{P}$, genotype k may belong to two subsets S_i and S_j. Without loss of generality, assume $g_{kp} = 0$ and $g_{ip} = 1$. Since haplotype h_i is associated to the subset S_i and explains genotype i then it must have the p-th SNP equal to 1 otherwise the sum operator would be violated. In turn, this implies that genotype k cannot be resolved by h_i since a necessary condition for its resolution is that the p-th SNP of h_i be equal to 0. Hence, independently of j, $y_{ij}^k = 0$ in any feasible solution of PPH. □

Finally, a similar process of reduction can be applied also to variables z_{ip} both by removing those whose value is fixed by constraints (6-7) and by the set of redundant variables (17-18). In this way only variables z_{ip} involved in constraints (8) need to be defined. In conclusion, denoted Y as the set of variables y_{ij}^k and $R = \cup_q R_q$, BM can be reduced to the following:

Formulation 2. - *Reduced Model (RM)*

$$\min \sum_{i \in K \cup K'} x_i \tag{19}$$

$$s.t.\ x_{i'} \leq x_i \qquad \forall\ i \in K \tag{20}$$

$$\sum_{i,j: y_{ij}^k \in Y \setminus R} y_{ij}^k \geq 1 \qquad \forall\ k \in K \tag{21}$$

$$\sum_{j: y_{ij}^k \in Y \setminus R, j \geq i} y_{ij}^k + \sum_{j: y_{ij}^k \in Y \setminus R, j < i} y_{ji}^k \leq x_i \qquad \forall\ k \in K, \forall\ i : y_{ij}^k \in Y \setminus R \tag{22}$$

$$y_{kk'}^k \leq x_{k'} \qquad \forall\ k \in K \tag{23}$$

$$z_{kp} + z_{k'p} = 1 \qquad \forall\ k \in K, \forall\ p \in \mathcal{P} : g_{kp} = 2 \tag{24}$$

$$z_{ip} \leq 1 - \sum_{j: y_{ij}^k \in Y \setminus R, j \geq i} y_{ij}^k - \sum_{j: y_{ij}^k \in Y \setminus R, j < i} y_{ji}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i : y_{ij}^k \in Y \setminus R, g_{kp} = 0, g_{ip} = 2 \tag{25}$$

$$z_{ip} \geq \sum_{j: y_{ij}^k \in Y \setminus R, j \geq i} y_{ij}^k + \sum_{j: y_{ij}^k \in Y \setminus R, j < i} y_{ji}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i : y_{ij}^k \in Y \setminus R, g_{kp} = 1, g_{ip} = 2 \tag{26}$$

$$z_{ip} \geq y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i,j : y_{ij}^k \in Y \setminus R, g_{ip} = 2, g_{jp} = 0, g_{kp} = 2 \tag{27}$$

$$z_{jp} \geq y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i,j : y_{ij}^k \in Y \setminus R, g_{ip} = 0, g_{jp} = 2, g_{kp} = 2 \tag{28}$$

$$z_{ip} \leq 1 - y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i,j : y_{ij}^k \in Y \setminus R, g_{ip} = 2, g_{jp} = 1, g_{kp} = 2 \tag{29}$$

$$z_{jp} \leq 1 - y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i,j : y_{ij}^k \in Y \setminus R, g_{ip} = 1, g_{jp} = 2, g_{kp} = 2 \tag{30}$$

$$z_{ip} + z_{jp} \geq y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i,j : y_{ij}^k \in Y \setminus R, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2 \tag{31}$$

$$z_{ip} + z_{jp} \leq 2 - y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ k \in K, \forall\ i,j : y_{ij}^k \in Y \setminus R, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2 \tag{32}$$

$$x_i,\ z_{ip},\ y_{ij}^k \in \{0,1\}. \tag{33}$$

Note that the integrality of variables y_{ij}^k suffices to guarantee the integrality of variables x_i. Hence, we denote by RMM the formulation in which the integrality condition on variables x_i is related.

3.3 Strengthening Inequalities

In this section we provide valid inequalities to strengthen RM.

Proposition 3. *The inequality*

$$\sum_{k \in K: y_{ij}^k \in Y \setminus R} y_{ij}^k \leq x_i \qquad \forall\ i,j : \exists\ y_{ij}^k \in Y \setminus R \tag{34}$$

is valid for RM.

Proof. Two subsets S_i and S_j a of feasible solution can share at most one genotype. Then, it follows that $\sum_{k \in K} y_{ij}^k \in \{0,1\}$. If the solution is such that $\sum_{k \in K} y_{ij}^k = 0$ then the inequality reduces to $x_i \geq 0$ which is trivially valid. If the solution is such that $\sum_{k \in K} y_{ij}^k = 1$ then genotype k belongs to the subset represented by genotype i which in turn implies $x_i = 1$, and the inequality is again valid. □

Similarly, it easy to see that the following proposition holds:

Proposition 4. *The inequality*

$$\sum_{k \in K: y_{ij}^k \in Y \setminus R} y_{ij}^k \leq x_j \qquad \forall\ i,j : \exists\ y_{ij}^k \in Y \setminus R \tag{35}$$

is valid for RM.

Note that $\sum_{k\in K} y_{ij}^k \in \{0,1\}$ also implies $\sum_{k\in S} y_{ij}^k \in \{0,1\}$, for any $S \subseteq K$. Let the set of genotypes $\mathcal{G}_p^1 = \{g_k,\ k \in K : g_{kp} = 1\}$. The following proposition holds:

Proposition 5. *The inequality*

$$z_{ip} \geq \sum_{k\in\mathcal{G}_p^1 : y_{ij}^k \in Y\setminus R} y_{ij}^k \qquad \forall p \in \mathcal{P},\ \forall\ i,j : y_{ij}^k \in Y \setminus R \tag{36}$$

is valid for RM.

Proof. If the solution is such that $\sum_{k\in\mathcal{G}_p^1} y_{ij}^k = 0$ then the inequality reduces to $z_{ip} \geq 0$ which is trivially valid. If the solution is such that $\sum_{k\in\mathcal{G}_p^1} y_{ij}^k = 1$ then the inequality reduces to $z_{ip} = 1$ which is again valid. □

Note that the inequality (36) only applies if $g_{ip} = 2$. In fact, if $g_{ip} = 1$ then (36) is redundant, and if $g_{ip} = 0$ then the corresponding variables y_{ij}^k are not defined.

By analogy, fixed i, j, and p, and considered the set of genotypes $\mathcal{G}_p^0 = \{g_k,\ k \in K : g_{kp} = 0\}$, the following proposition holds:

Proposition 6. *The inequality*

$$z_{ip} \leq x_i - \sum_{k\in\mathcal{G}_p^0 : y_{ij}^k \in Y\setminus R} y_{ij}^k \qquad \forall p \in \mathcal{P},\ \forall\ i,j : \exists\ y_{ij}^k \in Y \setminus R \tag{37}$$

is valid for RM.

Proof. If the solution is such that $\sum_{k\in\mathcal{G}_p^0} y_{ij}^k = 0$ then the inequality reduces to $z_{ip} \leq x_i$ which is trivially valid. If the solution is such that $\sum_{k\in\mathcal{G}_p^0} y_{ij}^k = 1$ then genotype k belongs to the subset represented by genotype i which implies $x_i = 1$ and thus $z_{ip} = 0$. □

Also in this case, the inequality (37) only applies if $g_{ip} = 2$. In fact, if $g_{ip} = 0$ then (37) is redundant, and if $g_{ip} = 1$ then the corresponding variables y_{ij}^k are not defined.

Defined $\mathcal{G}_p^2 = \{g_k : g_{kp} = 2,\ \forall\ k \in K\}$, the following proposition holds:

Proposition 7. *The inequality*

$$z_{ip} + z_{jp} \geq \sum_{k\in\mathcal{G}_p^2 : y_{ij}^k \in Y\setminus R} y_{ij}^k + 2 \sum_{k\in\mathcal{G}_p^1 : y_{ij}^k \in Y\setminus R} y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ i,j : \exists\ y_{ij}^k \in Y \setminus R \tag{38}$$

is valid for RM.

Proof. The left-hand-side $\sum_{k\in\mathcal{G}_p^2} y_{ij}^k + 2\sum_{k\in\mathcal{G}_p^1} y_{ij}^k \in \{0,1,2\}$. If the solution is such that $\sum_{k\in\mathcal{G}_p^2} y_{ij}^k + 2\sum_{k\in\mathcal{G}_p^1} y_{ij}^k = 0$ then the inequality reduces to $z_{ip} + z_{jp} \geq 0$ which is trivially valid. If the solution is such that $\sum_{k\in\mathcal{G}_p^2} y_{ij}^k = 1$ then the inequality reduces to $z_{ip} + z_{jp} \geq 1$ which is still valid. Finally, if the solution is such that $\sum_{k\in\mathcal{G}_p^1} y_{ij}^k = 1$ then the inequality reduces to $z_{ip} + z_{jp} = 2$ which is again valid. In fact, if haplotypes h_i and h_j resolve genotype k then they must be characterized by having the p-th SNP equal to 1. □

Similar arguments can be used to prove the following propositions:

Proposition 8. *The inequality*

$$z_{ip} + z_{jp} \leq x_i + x_j - \sum_{k\in\mathcal{G}_p^2 : y_{ij}^k \in Y\setminus R} y_{ij}^k - 2 \sum_{k\in\mathcal{G}_p^0 : y_{ij}^k \in Y\setminus R} y_{ij}^k \qquad \forall\ p \in \mathcal{P}, \forall\ i,j : \exists\ y_{ij}^k \in Y \setminus R \tag{39}$$

is valid for RM.

Genotypes	SNPs	Recombination level	Mean number of 2s sites	Max number of 2s per genotype (Mean)	Number of datasets
40	50	random in $\{0, 4, 16\}$	204.60	13.80	10
70	70	random in $\{0, 4, 16\}$	579.90	17.31	10
100	100	random in $\{0, 4, 16\}$	1047.20	22.30	10

Table 2: Overview of the datasets analyzed to compare the Basic and the Reduced models.

A distinct class of inequalities can be obtained by introducing the concept of *conflict* among genotypes (Bertolazzi et al. (2008)). Specifically, consider a pair of two genotypes (g_i, g_j) and assume that there exists at p-th SNP a conflict among g_i and g_j, i.e., one of the two genotypes (e.g., g_i) is such that $g_{ip} = 1$ whereas the corresponding entry of the other is such that $g_{jp} = 0$ (or vice-versa). Then, this condition is sufficient to exclude common haplotypes explaining g_i, g_j since their existence would violate the haplotype sum operator. The rationale can be extended to all the subset of genotypes having conflicts, and can be formalized as follows.

Define CG as a graph whose set of vertices is the set of genotypes $\mathcal{G}$, and whose set of edges contains all the pair of genotypes (g_i, g_j) that are in conflict. Then, the following preposition hold:

Proposition 9. *Let C be a clique in CG, then the inequality*

$$\sum_{k \in C} \sum_{j : y_{ij}^k \in Y \setminus R, j \neq i} y_{ij}^k \leq x_i \qquad \forall\ i : y_{ij}^k \in Y \setminus R,\ \forall\ C\ \subseteq CG \tag{40}$$

is valid for RM.

Proof. Only one genotype in C can be shared by two subsets S_i and S_j a of feasible solution. This implies that $\sum_{k \in C} \sum_{j : y_{ij}^k \in Y \setminus R, j \neq i} y_{ij}^k \in \{0, 1\}$. If the solution is such that $\sum_{k \in C} \sum_{j : y_{ij}^k \in Y \setminus R, j \neq i} y_{ij}^k = 0$ then the inequality reduces to $x_i \geq 0$ which is trivially valid. If the solution is such that $\sum_{k \in C} \sum_{j : y_{ij}^k \in Y \setminus R, j \neq i} y_{ij}^k = 1$ then the genotype k belongs to the subset represented by genotype i which in turn implies $x_i = 1$, hence the inequality is again valid. □

Proposition 9 can be further extended to variables z_{ip}. Specifically, define the set $E_p^1 = \{(g_i, g_j) \in \mathcal{G}_p^1 :\ \exists\ q \in \mathcal{P},\ q \neq p,\ g_{iq} + g_{jq} = 1\}$, for all $p \in \mathcal{P}$, and consider the graph $CG_p^1 = (\mathcal{G}_p^1, E_p^1) \subset CG$. Then, the following proposition holds:

Proposition 10. *The following inequality*

$$z_{ip} \geq \sum_{k \in C^1} \sum_{j : y_{ij}^k \in Y \setminus R, j \geq i} y_{ij}^k + \sum_{k \in C^1} \sum_{j : y_{ij}^k \in Y \setminus R, j < i} y_{ji}^k \qquad \forall\ p \in \mathcal{P}, \forall\ i : y_{ij}^k \in Y \setminus R,\ \forall\ \textit{clique}\ C^1 \subseteq CG_p^1 \tag{41}$$

is valid for RM.

By analogy, defined the set $E_p^0 = \{(g_i, g_j) \in \mathcal{G}_p^0 :\ \exists\ q \in \mathcal{P},\ q \neq p,\ g_{iq} + g_{jq} = 1\}$, for all $p \in \mathcal{P}$, and considered the graph $CG_p^0 = (\mathcal{G}_p^0, E_p^0) \subset CG$, the following proposition holds:

Proposition 11. *The inequality*

$$z_{ip} \leq x_i - \sum_{k \in C^0} \sum_{j : y_{ij}^k \in Y \setminus R, j \geq i} y_{ij}^k - \sum_{k \in C^0} \sum_{j : y_{ij}^k \in Y \setminus R, j < i} y_{ji}^k \qquad \forall\ p \in \mathcal{P}, \forall\ i : y_{ij}^k \in Y \setminus R,\ \forall\ \textit{clique}\ C^0 \subseteq CG_p^0 \tag{42}$$

is valid for RM.

The proofs of Propositions 10 and 11 are quite similar to the ones for Propositions 7 and 8 and are omitted. Finally, observe that Propositions 10 and 11 imply Proposition 9 and only apply if $g_{ip} = 2$.

4 Experiments

In this section we analyze the performances of our model to solve the pure parsimony haplotyping problem. Our experiments have been motivated by a number of goals, namely: to evaluate, with respect to BM, the benefits obtained by removing the redundant variables and including the valid inequalities previously described; to compare the performances of our model with the ones obtained by Brown and Harrower' HybridIP model (Brown and Harrower (2006)), currently the best model for PPH; and finally, to allow the analysis of larger datasets with respect to the ones currently analyzed.

Similarly to Brown and Harrower (2006), we wish to emphasize that our experiments aim simply to evaluate the runtime performance of our model for solving PPH. We neither attempt to study the efficiency of PPH for haplotype inference nor compare our algorithm to haplotype inference solvers that do not use the parsimony criterion; this analysis has been already performed by both Gusfield (2003) and Wang and Xu (2003) and we defer the reader interested to their articles.

4.1 Implementation

We have implemented BM and RM by means of Mosel 2 of Xpress-MP, Optimizer version 18, running on a Pentium 4, 3.2 GHz, equipped with 2 GByte RAM and operating system Gentoo release 7 (kernel linux 2.6.17). We have combined RM with different constraints obtaining two other models, specifically: RM including the valid inequalities (hereafter indicated with SM), and RMM including the valid inequalities and such that variables x_i are declared continuous (hereafter indicated with SMM). In order to evaluate the benefits of the valid inequalities we have deactivated the Xpress Optimizer automatic cuts and the pre-solving strategy.

4.2 Separation Oracle for the Valid Inequalities

When using models SM and SMM, the valid inequalities (34) and (35) are loaded together with RM. On the contrary, the valid inequalities (36)-(39) are dynamically generated by means of a separation oracle. Specifically, let $(\bar{x}, \bar{y}, \bar{z})$ be a current solution at a given node of the search tree. For each $p \in \mathcal{P}$, we build the set $\mathcal{G}_p^1$ and for all i, j such that $\exists\ y_{ij}^k \in Y \setminus R$ we check if $\bar{z}_{ip} < \sum_{k \in \mathcal{G}_p^1 : y_{ij}^k \in Y \setminus R} \bar{y}_{ij}^k$. If so, the inequality

$$z_{ip} \geq \sum_{k \in \mathcal{G}_p^1 : y_{ij}^k \in Y \setminus R} y_{ij}^k$$

is added to the formulation. To reduce the computational overhead, no more than 10 cuts are added at each node of the search tree. A similar procedures is used to separate inequalities (37)-(39). If no inequality of type (36)-(39) is added to the formulation then the separation oracle check for violated clique inequalities (41)-(42). Before running the exact search, we pre-compute the conflict graph CG for the instance analyzed. Subsequently, at a given node of the search tree, we pick at random an index $p \in \mathcal{P}$ and an index $i : y_{ij}^k \in Y \setminus R$, we dynamically construct the graph CG_p^1, and set $\bar{w}_k := \sum_{j : y_{ij}^k \in Y \setminus R, j \geq i} \bar{y}_{ij}^k + \sum_{j : y_{ij}^k \in Y \setminus R, j < i} \bar{y}_{ji}^k$, $k \in CG_p^1$. If $\bar{z}_{ip} \geq \sum_{k \in \mathcal{G}_p^1} \bar{w}_k$ then there not exists any violated inequality in CG_p^1 and we select at random a different index i; otherwise we proceed by finding a maximum clique in CG_p^1. After testing alternative (heuristic and exact) solution strategies for finding a maximum clique in a graph, we opted for Nemhauser and Trotter's exact algorithm (Pardalos and Xue (1994)) which proved to be, for the datasets analyzed, a good trade-off between speed (a few milliseconds) and efficiency. Specifically, Nemhauser and Trotter's algorithm consists in solving the model

Formulation 3.

$$\max \quad z_o = \sum_{k=1}^{|\mathcal{G}_p^1|} \bar{w}_k t_k \tag{43}$$

$$s.t. \quad t_i + t_j \leq 1\ \forall\ (i,j) \in \bar{E}_p^1 \tag{44}$$

$$t_i \in \{0,1\} \tag{45}$$

Datasets	Time (s)		RAM required (MB)		Max # of branches		Fraction Solved	
	BM	RM	BM	RM	BM	RM	BM	RM
40x50s	20.572 ± 0.77	1.551 ± 0.078	325 ± 20	5 ± 1	1	1	10/10	10/10
70x70s	-	9.995 ± 0.94	> 1058	9 ± 3	-	1	0/10	10/10
100x100s	-	39.249 ± 2.33	> 1933	14 ± 6	-	1	0/10	10/10

Table 3: Results obtained by the Basic and the Reduced models on the datasets analyzed.

where t_i denotes a decision variable equal to 1 if vertex $t_i \in \mathcal{G}_p^1$ belongs to the clique and 0 otherwise, and $\bar{E}_p^1$ is the complement of the edge set E_p^1. If the optimal value of the objective function is greater than $\bar{z}_{ip}$ then the inequality

$$z_{ip} \geq \sum_{k \in C^1} \sum_{j : y_{ij}^k \in Y \setminus R, j \geq i} y_{ij}^k + \sum_{k \in C^1} \sum_{j : y_{ij}^k \in Y \setminus R, j < i} y_{ji}^k$$

is added to the formulation. A similar procedure is used for separating the clique inequalities (42). In order to speed up the cut generation, branching priorities on the integer variables have been introduced. Specifically, first the branches are performed on variables z_{ip}, then on variables x_i and finally on variables y_{ij}^k. Moreover, the search is performed before on variables z_{ip} whose relaxed values are in the interval $(0, 0.5]$ and subsequently on those whose relaxed values is in the interval $(0.5, 1)$.

4.3 Basic and Reduced Models on Simulated Datasets

Analogously to Brown *et al.* (Brown and Harrower (2006)), we used the Hudson's MS program (Hudson (1990)) to generate benchmark simulated datasets in order to compare the performances of BM and RM. Under the neutral evolution hypothesis (Kimura (1983)), MS is able to generate many independent replicate samples (haplotypes) through a variety of assumptions about migration, recombination rate and population size. A coalescent approach (Hudson (1990)) is at the core of the simulator: a random genealogy of the sample is firstly generated, and subsequently mutations are randomly placed on the genealogy. A parameter called recombination level allows one to tune the heterogeneity degree of the haplotypes in a population: as r increases, haplotypes become more different from each others, and consequently the number of heterozygous sites in genotype samples tends to become larger.

Three sets of samples were generated, specifically: 10 datasets containing 80 samples of 50 SNPs each, 10 datasets containing 140 samples of 70 SNPs each, and 10 datasets containing 200 samples of 100 SNPs each, respectively. For each dataset, the recombination level was randomly chosen in $\{0, 4, 16\}$. In order to obtain datasets of genotypes, sample datasets were paired up in a random way such that all of them were used. This process led to the generation of three datasets of 10 instances each containing 40 genotypes, 70 genotypes and 100 genotypes, respectively. Duplicate genotypes were removed though a preprocessing step. It is worth noting that this reduction does not interfere with the value of the optimal solution, and reduce the gap between the optimal solution of the IP problem and its relaxation. Furthermore, Brown and Harrower's preprocessing step consisting in the elimination of duplicate columns was also implemented (Brown and Harrower (2006)).

Table 2 summarizes the details about the genotype datasets used, whereas Table 3 shows the results obtained by BM and RM on the datasets analyzed, specifically: the second and third columns of Table 3 indicates the mean and the standard deviation of the time (RAM) required by BM and RM to solve a given dataset. The fourth column indicates the maximum number of branch-and-bound branches required by BM and RM to solve a given dataset. Finally the fifth column shows the overall number of instances solved by BM and RM on each dataset.

As general trend, numerical experiments show that RM performs better than BM both in terms of runtime and memory. The reduction of variables y_{ij}^k and z_{ip} (and the corresponding constraints) performed by RM on the datasets analyzed approaches the 99% of the overall number of variables and constraints required by BM. As consequence, this strategy leads to a model requiring only a few MBytes RAM, whereas several hundreds of MBytes RAM are required by the BM model (main reason for which the BM model is unable to tackle instances of PPH bigger than 40 genotypes).

Due to its better performance, we have preferred to use RM in place of BM for the further numerical analysis described in the next section.

Uniform				
Genotypes	SNPs	Max # of 1s in a genotype	Mean # of 1s in a genotype	Recombination level
50	10	8	3.1	0
50	10	9	3.3	4
50	10	9	2.9	16
50	30	23	8.2	0
30	50	42	14.1	0
30	75	52	19.8	0
30	100	68	25.1	0
Nonuniform				
Genotypes	SNPs	Max # of 1s in a genotype	Mean # of 1s in a genotype	Recombination level
50	10	7	1.7	n.a.
50	30	18	5.0	n.a.
50	50	33	8.4	n.a.
30	75	41	15.8	n.a.
30	100	66	19.1	n.a.

Table 4: Characteristics of Brown and Harrower's artificial datasets.

4.4 Performance Analysis on Brown and Harrower's Datasets

We have used Brown and Harrower's datasets (Brown and Harrower (2006)) for testing the performances of our models versus Brown and Harrower' HybridIP state-of-the-art model for PPH. Specifically, through the Hudson's MS program (Hudson (1990)), the authors created two families of datasets (called the *uniform* and *nonuniform* datasets) by randomly pairing the resulting haplotypes. The distinction in the two simulated methods comes in how the random pairing is performed. In the uniform datasets the haplotypes are randomly paired by sampling uniformly from the set of distinct haplotypes. In the nonuniform datasets the haplotypes are sampled uniformly from the collection of haplotypes generated by the coalescent process. In this collection, haplotypes may not be unique, so some haplotypes are sampled with higher frequency than others. The uniform datasets consist of collections of 50 genotypes with haplotypes having either 10 or 30 SNPs. The nonuniform datasets consist of collections of either 30 or 50 haplotypes of length 50, 75, and 100. Each dataset contains a number of instances variable between 15 and 50. Table 4 resumes the main characteristics of Brown and Harrower's artificial datasets.

Brown and Harrower also considered biological inputs from chromosomes 10 and 21, over all four Hap-Map (Consortium (2004)) populations. For each input length the authors selected sequences of lengths 30, 50, and 75, giving a total of 8 datasets consisting of 3 instances each. Table 5 resumes the main characteristics of Brown and Harrower's biological datasets.

We show in Tables 6-8 the performances of our models on the uniform, nonuniform and biological datasets, respectively. Specifically, the columns of each table evidence the mean, the maximum, and the minimum of: the solution time, the gap (i.e., the difference between the optimal value found and the value of linear relaxation at the root node of the search tree, divided by the optimal value), the number of branches performed, and the number of cuts added to solve each group of instances belonging to a given dataset.

4.4.1 Uniform Datasets

Brown and Harrower (2006) showed that the HybridIP model is able to solve instances of the datasets having a genotype length 30 or less. Specifically, the authors observed that HybridIP needs a runtime ranging from 7 seconds to 2 minutes to solve instances of dataset having genotype length 10, and from 30 seconds to an hour to solve instances of the dataset having genotype length 30. However, HybridIP is only partially able to solve instances of the datasets having a genotype length 30 or greater. In fact, HybridIP solved the 70 percent of the instances of the dataset having a genotype length 50, the 60 percent of instances of the dataset having a genotype length 75, and only the 30 percent of the instances of the dataset having a genotype length 100. The runtimes range from 5 seconds to two hours.

On the contrary, our numerical experiments (see Table 6) show that RM in average is able to solve instances of the datasets having genotype length 10 in about 8 seconds, although five instances of the dataset

50x10r16 (specifically, instances 02, 04, 06, 11 and 13) took a bit longer (22.44, 11.152, 30.623, 16.535, and 11.822 seconds, respectively). Instances of the dataset having genotype length 30 in average have been solved in about 11.772 seconds, and two instances (02 and 08) took more than the average (53.42 and 38.642 seconds respectively), whereas the remaining one took less than 9 seconds. Instances of the datasets having genotype length greater than 30 in average have been solved in at most 15.624 seconds, with a maximum solution time equal to 47.467, hence much less than HybridIP. It is worth noting that the gap of each dataset is 0 in at least the 50% of the instances analyzed and that several instances are typically solved without branching.

SM shows better performances then RM (see Table 7) both in terms of solution times and gaps. In fact, in average, SM is able to solve instances of the datasets having genotype length 10 in about 5.8 seconds, with maximum and minimum solution times always smaller than RM. The first two datasets have been solved at root node whereas an instance of the third dataset needed at most of 49 branches to reach the optimum. The efficiency of the valid inequalities is confirmed by both the smaller gaps than RM and the average number of cuts added during the exact search. This trend is confirmed also when the analyzing datasets having genotype length greater or equal to 30, which the only exception of the instance 19 of the dataset 30x50in which took 82.677 seconds, against 25.791 seconds of RM. Specifically, this instance is characterized by a gap of 2.63158 (against 0.0 of RM), a number of branches equal to 1109 (against 75 of RM) and a number of cuts added equal to 741. The anomaly of this instance seems to be due to the valid inequalities (34) and (35) which negatively interfere with the primal heuristic of the Xpress Optimizer.

Finally, SMM shows worse running time performances than RM and SM (see Table 8) in all uniform datasets. This behavior is in contrast with the the average, maximum and minimum gap which is always smaller than RM and SM. Once again, the anomalous behavior is due to the valid inequalities (34) and (35) which interfere both with the primal heuristic of the Xpress Optimizer and with the automatic branching strategy.

			Biological	
Dataset	Genotypes	SNPs	Max # of 1s in a genotype	Mean # of 1s in a genotype
test-chr10-CEU-30	36	30	19	0.402778
test-chr10-CEU-50	36	50	24	0.235
test-chr10-CEU-75	36	75	39	0.166667
test-chr10-HCB-30	20	30	17	0.271667
test-chr10-HCB-50	20	50	25	0.11
test-chr10-HCB-75	20	75	41	0.238667
test-chr10-JPT-30	11	30	13	0.251515
test-chr10-JPT-50	11	50	27	0.356364
test-chr10-JPT-75	11	75	40	0.437576
test-chr10-YRI-30	33	30	20	0.243434
test-chr10-YRI-50	33	50	29	0.365455
test-chr10-YRI-75	33	75	47	0.452121
test-chr10-CEU-30	32	30	19	0.427083
test-chr10-CEU-50	32	50	24	0.264375
test-chr10-CEU-75	32	75	39	0.1875
test-chr10-HCB-30	7	30	17	0.447619
test-chr10-HCB-50	7	50	25	0.297143
test-chr10-HCB-75	7	75	41	0.502857
test-chr10-JPT-30	15	30	13	0.184444
test-chr10-JPT-50	15	50	27	0.261333
test-chr10-JPT-75	15	75	40	0.355556
test-chr10-YRI-30	68	30	20	0.118137
test-chr10-YRI-50	68	50	29	0.177353
test-chr10-YRI-75	68	75	47	0.260392

Table 5: Characteristics of Brown and Harrower's biological datasets.

RM

Dataset	Time (sec.)			Gap (%)			Nodes		
	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform									
50x10	1.143	2.404	0.102	0.000	0	0	1.000	1	1
50x10r4	1.730	6.104	0.043	1.179	10	0	1.000	1	1
50x10r16	8.092	30.623	2.011	1.644	10.7692	0	1.533	9	1
50x30	11.772	53.42	2.732	2.440	7.14286	0	2.000	15	1
30x50in	8.922	47.467	0.73	1.694	7.69231	0	10.260	75	1
30x75in	15.624	35.693	1.358	1.649	6.66667	0	24.300	92	1
30x100in	10.142	31.994	2.593	1.402	7.35294	0	8.500	25	1
Nonuniform									
50x10	0.634	1.726	0.127	0.513	7.69231	0	2.400	11	1
50x30	11.882	30.411	1.59	1.164	6.25	0	11.867	35	1
30x50in	10.764	24.108	0.815	0.890	4.09091	0	20.533	61	1
30x75in	22.389	61.869	3.537	1.038	5.55556	0	62.286	387	1
30x100in	74.925	462.791	12.953	1.521	4.7619	0	216.071	1679	8
Biological									
CHR10-CEU	102.792	305.103	0.774	0.000	0	0	270.333	807	1
CHR10-HCB	38.058	96.324	8.746	2.593	7.77778	0	67.000	151	1
CHR10-JPT	0.895	1.583	0.368	1.515	4.54545	0	7.000	11	1
CHR10-YRI	73.723	116.127	31.353	1.111	3.33333	0	89.667	123	63
CHR21-CEU	18.868	54.562	0.428	1.515	4.54545	0	49.667	145	1
CHR21-HCB	0.182	0.456	0.017	0.000	0	0	8.000	19	1
CHR21-JPT	1.781	2.87	0.967	0.833	2.5	0	15.667	29	1
CHR21-YRI	2349.331	6819.2	50.012	0.000	0	0	3815.667	11199	123

Table 6: Performances of RM.

SM

Dataset	Time (sec.)			Gap (%)			Nodes			Cuts		
	Average	Max	Min	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform												
50x10	0.835	2.339	0.085	0.000	0	0	1.000	1	1	0.667	10	0
50x10r4	1.350	3.269	0.023	0.667	10	0	1.000	1	1	2.000	10	0
50x10r16	5.848	21.293	1.611	1.201	5.55556	0	4.733	49	1	4.600	20	0
50x30	8.305	32.291	1.627	1.609	7.14286	0	5.267	34	1	8.667	32	0
30x50in	6.227	82.677	0.366	0.934	9.09091	0	48.240	1109	1	37.080	741	0
30x75in	10.836	20.595	1.519	0.979	6.66667	0	49.900	175	1	36.900	116	10
30x100in	9.515	27.67	2.413	1.305	6.66667	0	25.800	103	1	23.400	69	10
Nonuniform												
50x10	0.492	0.956	0.086	0.513	7.69231	0	2.800	9	1	6.933	17	0
50x30	7.780	24.36	0.88	0.556	5.55556	0	24.867	77	1	21.667	58	0
30x50in	10.553	38.569	0.754	0.540	5.15873	0	110.667	559	1	62.867	203	0
30x75in	27.939	92.472	2.888	0.722	5.55556	0	282.786	1477	1	138.571	569	10
30x100in	126.111	7227.52	5.798	0.000	3.99743	0	2434.000	94846	34	1748.857	31915	36
Biological												
CHR10-CEU	46.759	138.315	0.282	0.000	0	0	137.000	401	1	50.667	142	0
CHR10-HCB	1026.015	3067.22	5.362	1.754	5.26316	0	11041.667	32981	29	9948.333	29760	20
CHR10-JPT	0.977	2.453	0.124	0.000	0	0	61.667	173	6	46.000	104	17
CHR10-YRI	73.674	135.232	38.146	0.617	1.85185	0	199.000	253	141	97.000	125	73
CHR21-CEU	449.307	1346.36	0.227	1.515	4.54545	0	4289.000	12857	1	1810.333	5419	0
CHR21-HCB	0.096	0.2	0.008	0.000	0	0	2.667	4	1	9.333	16	0
CHR21-JPT	2.297	4.42	0.085	0.282	3.84615	0	85.667	209	1	65.333	152	0
CHR21-YRI	95.334	160.563	30.105	0.000	0	0	205.000	269	141	112.000	131	93

Table 7: Performances of SM.

4.4.2 Nonuniform Datasets

Brown and Harrower (2006) showed that in the nonuniform datasets the HybridIP solves the instances having length 10 and 30. Specifically, the instances having genotypes length 10 are solved in one second, and 10 out of the 15 length 30 instances solved in less than 15 seconds. However, the HybridIP is able to solve only six instances having genotype length 50, although those instances have been solved in 39 seconds or less. Again, the Hybrid is able to solve 5 of the 15 instances with genotype length 75, thereof two of those are solved in less than one minute. Finally, the HybridIP is able to solve 3 instances having genotype length 100, thereof two of those solved in under two minutes.

Numerical experiments have shown that our models outperforms HybridIP also in the nonuniform datasets. In fact, as shown in Table 6, RM in average is able to solve instances of the datasets having genotype length 10 in about 0.634 seconds, with a maximum runtime of 1.726 seconds; instances having genotype length 30 in about 11.882 seconds, with a maximum runtime of 30.411 seconds; instances having genotype length 50 in about 10.764 seconds, with a maximum runtime of 24.108 seconds; instances having genotype length 75 in about 22.389 seconds, with a maximum runtime of 61.869 seconds; and finally, instances having genotype

SMM

Dataset	Time (sec.)			Gap (%)			Nodes			Cuts		
	Average	Max	Min	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform												
50x10	1.139	3.383	0.088	0.000	0	0	1.667	4	1	3.200	13	0
50x10r4	3.306	17.695	0.021	0.000	0	0	2.933	16	1	7.333	17	0
50x10r16	22.720	102.463	3.864	0.000	0	0	17.800	89	1	16.133	31	0
50x30	35.562	152.909	1.285	0.000	0	0	18.867	61	1	17.600	44	0
30x50in	23.860	415.269	0.787	0.121	3.125	0	93.900	2399	1	67.260	1736	0
30x75in	24.163	85.899	1.935	0.000	0	0	95.300	369	1	70.100	259	13
30x100in	15.769	47.697	4.1	0.000	0	0	42.100	223	1	29.900	129	11
Nonuniform												
50x10	0.976	1.611	0.251	0.000	0	0	5.733	19	1	11.400	20	0
50x30	22.448	78.588	2.02	0.196	2.94118	0	62.133	203	7	37.267	93	13
30x50in	34.690	266.58	2.919	0.144	1.85185	0	311.933	3367	9	138.000	1227	14
30x75in	81.194	445.243	7.713	0.054	0.757576	0	685.643	4063	21	413.429	1910	21
30x100in	658.337	7324.47	6.603	0.000	1.333	0	11124.214	116887	14	9114.571	101623	22
Biological												
CHR10-CEU	104.346	303.915	0.777	0.000	0	0	224.667	649	8	86.333	226	15
CHR10-HCB	1089.757	3245.86	7.693	0.292	0.877193	0	9190.333	27391	77	8411.333	25112	36
CHR10-JPT	1.910	4.07	0.319	0.000	0	0	54.000	135	11	44.000	87	20
CHR10-YRI	191.857	464.219	39.36	0.000	0	0	287.333	401	179	176.667	299	71
CHR21-CEU	892.729	2670.99	0.662	0.000	0	0	4376.000	13103	8	1771.333	5277	18
CHR21-HCB	0.187	0.411	0.008	0.000	0	0	6.333	13	1	15.333	23	10
CHR21-JPT	6.327	13.354	0.865	0.000	0	0	183.333	463	8	126.000	285	18
CHR21-YRI	206.543	7242.43	84.659	0.000	0	0	346.667	6204	401	221.333	2814	299

Table 8: Performances of SMM.

length 100 in about 74.925 seconds, with a maximum runtime of 462.791 seconds. SM performs better than RM when solving instances having length 50 or smaller, but results in average slower when analyzing greater instances. Specifically, SM is always faster than RM in almost all instances of the dataset 30x75in, with the only exception of the instances 02 and 14 whose solution time was 85.551 and 92.472 seconds, respectively. Similarly, SM is always faster than RM in almost all instances of the dataset 30x100in, with the only exception of the instances 01, 09 and 14 whose solution time was 267.156, 663.219 and 327.615 seconds, respectively. Moreover, SM was unable to solve within the limit time the instance 00 (it's solution time was 7227.52).

Finally, SMM shows worse running time performances than RM and SM (see Table 8) in all nonuniform datasets. It is worth noting that SMM is characterized by the same difficulties that SM when dealing with the instances 02 and 14 of the dataset 30x75in, and 00, 01, 09 and 14 of the dataset 30x100in (although the gaps are always better than the corresponding ones of RM and SM).

4.4.3 Biological Datasets

To complete the performance analysis on Brown and Harrower's datasets (Brown and Harrower (2006)) we tested our model on the biological datasets. Brown and Harrower showed that HybridIP is able to solve 16 of the 24 biological instances. Unfortunately, the runtimes for this dataset have not be provided by the authors.

As general trend the numerical experiments show that also on the biological datasets SM is generally the model characterized by better performances with the exception of the datasets CHR10-HCB, CHR21-CEU, and CHR21-JPT whose instances chr10-HCB-75, chr10-CEU-30 and chr10-JPT-30 took 3067.22, 1346.36 and 4.42 seconds, respectively, to be solved exactly. Possibly, the implementation of a primal heuristic may turn out helpful for decreasing the solution time of those instances in which the valid inequalities (34) and (35) negatively interfere with the Xpress Optimizer primal heuristic. However, the benefits of the valid inequalities are evident when analyzing e.g., the datasets CHR10-CEU and CHR21-YRI: specifically, in the former case the solution time is halved, and in the latter case the solution time for the most difficult instance decreases from almost two hours to almost three minutes.

4.5 Performance Analysis on Larger Datasets

We have generated new classes of datasets with the aim of studying the performances of our models on larger datasets than Brown and Harrower's ones (Brown and Harrower (2006)). Specifically, under the same sample generation described in Section 4.3, we have generated five classes of datasets characterized by 50, 75, 100, 200 and 300 genotypes each. Each class contains 8 instances characterized by a SNP length ranging between 100 and 800. We have fixed the runtime limit to two hours and considered as measure of the performance the time (in seconds) necessary to solve each instance. The results, listed Table 9, show that SM is characterized

Model	Genotypes	SNP length							
		100	200	300	400	500	600	700	800
RM	50	21.539	47.546	70.814	95.420	119.308	141.682	160.54	184.586
RM	75	76.530	145.127	225.173	314.501	396.974	457.972	560.465	615.152
RM	100	189.736	353.716	544.839	765.423	962.542	1119.63	927.976	731.711
RM	200	845.646	1648.66	5237.8	3539.14	5355.9	5443.56	6244.8	-
RM	300	3944.94	-	-	-	-	-	-	-
SM	50	14.671	19.974	26.511	40.017	62.375	77.621	98.120	112.353
SM	75	41.658	94.860	113.958	171.644	281.06	217.39	391.29	490.83
SM	100	104.031	296.637	259.112	399.288	610.626	532.14	771.02	541.03
SM	200	625.018	1118.448	3665.77	1608.02	4019.28	-	-	-
SM	300	-	-	-	-	-	-	-	-
SMM	50	32.552	65.412	115.869	156.771	131.420	176.008	212.697	244.881
SMM	75	108.810	271.341	191.23	398.29	503.754	531.244	571.221	799.84
SMM	100	265.200	361.299	600.554	846.443	1207.97	1757.992	1727.02	1047.39
SMM	200	1315.935	1706.01	6923.9	4760.1	7144.45	-	-	-
SMM	300	-	-	-	-	-	-	-	-

Table 9: Performances of the models as function of the number of genotypes and SNP length.

by better runtime performances, although it is unable (together with SMM) to analyze datasets containing more than 300 genotypes and 600 SNPs. Only RM is able to analyze datasets containing more than 300 genotypes and 600 SNPs. This fact is due to the overhead of RAM introduced by the separation algorithm. Possibly, a combined use of the Xpress automatic cuts and pre-solving with a fine tuning on the number of valid inequalities added by the separation oracle at each node of the search tree could allow SM to analyze bigger datasets.

5 Conclusion

In this paper we have described a new polynomial formulation for the Pure Parsimony Haplotyping (PPH) problem. The idea at its core is that a solution for PPH induces a covering of the genotypes by subsets such that: (i) each subset of genotypes shares one haplotype, (ii) each genotype belongs to exactly two subsets, and (iii) every pair of subsets intersects in at most one genotype. This observation led us to develop a new integer linear model for PPH based on the class representatives with smallest index, already proposed for the coloring problem (Campelo et al. (2008)). We have suggested techniques to reduce the overall number of variables and constraints required by our models and provided valid inequalities. Computational experience showed that, under the same sample generation conditions used by Gusfield (2003), Brown and Harrower (2006), and Bertolazzi et al. (2008), our models outperform all existing IP models for PPH, whose limit is represented by datasets characterized by 68 genotypes of 75 SNPs each (Brown and Harrower (2006)). Our model can be applied to real biological genotype datasets much bigger than the ones proposed by (Brown and Harrower (2006)), and specifically characterized by hundreds of genotypes and SNPs. The model is compact, polynomial-sized, easy to implement, solvable with standard solvers, and usable in those cases for which the parsimony principle is well suited for haplotyping inference.

6 Acknowledgements

Daniele Catanzaro acknowledges support from the Belgian National Fund for Scientific Research (F.N.R.S.), of which he is a Research Fellow. Daniele Catanzaro and Martine Labbé also acknowledge support from Communauté Française de Belgique — Actions de Recherche Concertées (ARC). Finally, the authors thank Daniel G. Brown and Ian M. Harrower who kindly provided the benchmark datasets.

References

V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. *Journal of Computational Biology*, 10:323–340, 2003.

P. Bertolazzi, A. Godi, M. Labbé, and L. Tininini. Solving haplotyping inference parsimony problem using a new basic polynomial formulation. *Computers and Mathematics with Applications*, 55(5):900–911, 2008.

P. Bonizzoni, G. Della Vedova, R. Dondi, and L. Jing. The haplotyping problem: A view of computational models and solutions. *International Journal of Computers and Science Technology*, 18(6):675–688, 2003.

D. Brown and I. M. Harrower. A new integer programming formulation for the pure parsimony problem in haplotype analysis. In I. Jonassen and J. Kim, editors, *Proceedings of the Fourth Annual Workshop Algorithms in Bioinformatics*, volume 3240, pages 254–265. Springer-Verlag, Berlin, Germany, 2004.

D. Brown and I. M. Harrower. Integer programming approaches to haplotype inference by pure parsimony. *IEEE Transactions, Computational Biology and Bioinformatics*, 3(2):141–154, 2006.

M. Campelo, V. Campos, and R. Correa. On the asymmetric representatives formulation for the vertex coloring problem. *Discrete Applied Mathematics*, 156(7):1097–1111, 2008.

A. Chakravarti. It's raining snp, hallelujah? *Nature Genetics*, 19:216–217, 1998.

A. G. Clark, K. Weiss, D. Nickerson, S. Taylor, A. Buchanan, J. Stengard, V. Salomaa, E. Vartiainen, M. Perola, E. Boerwinkle, and C. Sing. Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase. *The American Journal of Human Genetics*, 63: 595–612, 1998.

The Int'l HapMap Consortium. Integrating ethics and science in the international hapmap project. *Nature Reviews Genetics*, 5(6):467–475, 2004.

E. Eskin, E. Halperin, and R. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of Bioinformatics and Computational Biology*, 1:1–20, 2003.

L. Excoffier and M. Slatkin. Maximum likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–927, 1995.

D. Fallin and N. J. Schork. Accuracy of haplotype frequency estimation for biallelic loci via the expectation maximization algorithm for unphased diploid genotype data. *American Journal of Human Genetics*, 67: 947–959, 2000.

M. R. Garey and D. S. Johnson. *Computers and Intractability.* Freeman, NY, USA, 2003.

D. Gusfield. Haplotype inference by pure parsimony. In Lecture Note in Computer Science, editor, *Annual Symposium in Combinatorial Pattern Matching*, volume 2676, pages 144–155. Springer-Verlag, Berlin, Germany, 2003.

L. Helmuth. Genome research: Map of the human genome 3.0. *Science*, 293:583–585, 2001.

M. Hoehe, K. Kopke, B. Wendel, K. Rohde, C. Flachmeier, K. Kidd, W. Berrettini, and G. Church. Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence. *Human Molecular Genetics*, 9:2895–2908, 2000.

R. R. Hudson. Gene genealogies and the coalescent process. *Oxford Survey of Evolutionary Biology*, 7:1–44, 1990.

M. Kimura. *The Neutral Theory of Molecular Evolution.* Cambridge University Press, Cambridge, U.K., 1983.

G. Lancia and R. Rizzi. A polynomial case of the parsimony haplotyping problem. *Operations Research Letters*, 34(3):289–295, 2006.

G. Lancia, M. C. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony: Complexity of exact and approximate algorithms. *INFORMS Journal on Computing*, 16(4):348–359, 2004.

E. Marshall. Drug firms to create public database of genetic mutations. *Science*, 284:406–407, 1999.

T. Niu, Z. S. Qin, and J. S. Liu. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *American Journal of Human Genetics*, 71:1242–1247, 2002a.

T. Niu, Z. S. Qin, X. Xu, and J. S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70:157–169, 2002b.

P. M. Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

R. Schwartz, A. G. Clark, and S.Istrail. Methods for inferring block-wise ancestral history from haploid sequences. In R. Guigo and D. Gusfield, editors, *Algorithms in Bioinformatics, Second International Workshop (WABI'02)*, volume 2452, pages 44–59. Springer-Verlag, Berlin, Germany, 2002.

C. Semple and M. Steel. *Phylogenetics.* Oxford University Press, NY, USA, 2003.

M. Stephens and P. Donnelly. A comparison of bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics*, 73:1162–1169, 2003.

M. Stephens, N. J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989, 2001.

J. Terwilliger and K. Weiss. Linkage disquilibrium mapping of complex disease: Fantasy and reality? *Current Opinions in Biotechnology*, 9:579–594, 1998.

J. C. Venter. The sequence of the human genome. *Science*, 291:1304–1351, 2001.

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