Université Libre de Bruxelles

Faculté des Science

MODELS AND METHODS IN GENOME WIDE ASSOCIATION STUDIES

LUCIANO PORRETTA

Service Graphes et Optimisation Mathématiques (G.O.M.)

Département d'Informatique

Promoteur:

Prof. Bernard Fortz

January 26, 2018

Thèse présentée en vue de l'obtention du grade académique de *Docteur en Sciences*

Université Libre de Bruxelles
Department of Computer Science
Graphes et Optimisation Mathématique
Boulevard du Triomphe CP 21001
B1050 Brussels Belgium

EMAIL:luciano.porretta@ulb.ac.be
URL:http://gom.ulb.ac.be/lporrett/

This thesis has been realised under the supervision of Prof. Bernard Fortz (ULB).

The members of the jury are:

- Prof. Daniele Catanzaro (Université Catholique de Louvain, Belgium)
- Prof. Bernard Fortz (Université Libre de Bruxelles, Belgium)
- Prof. Bjarni V. Haldòrsson (University of Reykjavìk, Iceland)
- Prof. Martine Labbé (Université Libre de Bruxelles, Belgium)
- Prof. Giuseppe Lancia (University of Udine, Italy)
- Prof. Tom Lenaerts (Université Libre de Bruxelles, Belgium)

SUBMISSION: 10 November 2017
PRIVATE DEFENSE: 12 December 2017
PUBLIC DEFENSE: 26 January 2018

"I have no special talents. I am only passionately curious"

—

Albert Einstein (11 March 1952)

ABSTRACT

The interdisciplinary field of systems biology has evolved rapidly over the last few years. Different disciplines have contributed to the development of both its experimental and theoretical branches. Although computational biology has been an increasing activity in computer science for more than a two decades, it has been only in the past few years that optimization models have been increasingly developed and analyzed by researchers whose primary background is *Operations Research* (OR). This dissertation aims at contributing to the field of computational biology by applying mathematical programming to certain problems in molecular biology. Specifically, we address three problems in the domain of *Genome-Wide Association Studies*: (i) the *Pure Parsimony Haplotyping under Uncertain Data Problem* that consists in finding the minimum number of haplotypes necessary to explain a given set of genotypes containing possible reading errors; (ii) the *Parsimonious Loss of Heterozygosity Problem* that consists of partitioning suspected polymorphisms from a set of individuals into a minimum number of deletion areas; (iii) and the *Multiple Individuals Polymorphic ALU Insertions Recognition Problem* that consists of finding the set of locations in the genome where ALU sequences are inserted in some individual(s). All three problems are *NP*-hard combinatorial optimization problems. Therefore, we analyse their combinatorial structure and we propose an exact approach to solution for each of them. The proposed models are efficient, accurate, compact, polynomial-sized and usable in all those cases for which the parsimony criterion is well suited for estimation.

RÉSUMÉ

Le domaine interdisciplinaire de la biologie des systèmes a évolué rapidement au cours des dernières années. Différentes disciplines ont contribué au développement de la branche expérimentale aussi bien que de la branche théorique Bien que la biologie computationnelle a été une activité en en croissance en informatique depuis plusde deux décennies, ce n'est que depuis quelques années que des modèles d'optimisation ont été de plus en plus développés et analysés par des chercheurs dont la spécialité de base est la recherche opérationnelle. Cette thèse vise à apporter une contribution au domaine de la biologie computationnelle en appliquant la programmation mathématique à certains problèmes de biologie moléculaire. Plus précisément, nous abordons trois problèmes dans le domaine de *Genome-Wide Association Studies*: (i) le problème appelé *Pure Parsimony Haplotyping under Uncertain Data*, qui consiste à trouver le nombre minimum d'haplotypes nécessaire pour expliquer un ensemble donné de génotypes contenant des erreurs de lecture potentielles; (ii) le problème appelé *Parsimonious Loss of Heterozygosity Problem*, qui consiste dans le partitionnement des polymorphismes soupçonnés à partir d'un ensemble d'individus en un nombre minimal de zones de suppression; (iii) et le troisième problème appelé *Multiple Individuals Polymorphic ALU Insertions Recognition Problem*, qui consiste à trouver l'ensemble des emplacements dans le génome où une séquence d'ALU est insérée dans certains individus. Les trois problèmes sont des problèmes d'optimisation combinatoire *NP*-difficile. Par conséquent, nous avons analysé leur structure combinatoire et proposé une approche exacte de résolution pour chacun d'entre eux. Les modèles proposés sont efficaces, précis, compacts, de taille polynomiale, et utilisables

dans tous les cas pour lesquels le critère de parcimonie est bien adapté à l'estimation.

PUBLICATIONS

Some chapters of this thesis are partially based on articles that have been published or submitted for publication.
Specifically:

Catanzaro, D., M. Labbé, and L. Porretta. 2011. "A Class Representative Model for Pure Parsimony Haplotyping under Uncertain Data." *PLOS ONE* 6, no. 3 ().

Porretta, L., D. Catanzaro, B. Fortz, and B. V. Haldòrsson. 2016. "A Branch&Price Algorithm for the Minimum Cost Clique Cover Problem in Max-Point Tolerance Graphs."

Porretta, L., B. Fortz, and B. V. Haldòrsson. 2017. "An Integer Programming formulation of the Multiple Individual Polymorphic ALU Insertion Recognition Problem."

"The isolated man does not develop any intellectual power. It is necessary for him to be immersed in an environment of other men, whose techniques he absorbs during the first twenty years of his life. He may then perhaps do a little research of his own and make a very few discoveries which are passed on to other men. From this point of view the search for new techniques must be regarded as carried out by the human community as a whole, rather than by individuals"

— Alan Turing

ACKNOWLEDGEMENTS

I want to convey my sincere gratitude to Professors Bernard Fortz, supervisor of this work, for offering me the possibility to write my thesis in G.O.M., for giving me the freedom to follow my research interests and finally for making sure that I could spend some of my time abroad.

I am very grateful to Professor Martine Labbé, who introduced me to the world of research during my master thesis and who encouraged me to pursue a Ph.D. degree in Brussels. I also want to thank the remaining members of my thesis committee for the time and effort they devoted to my work. In particular, I want to convey my gratitude to Professor Tom Lenaerts, who has been part of my *comité d'accompagnement* at ULB from the beginning of this thesis. I always appreciated his sincerity and moral support during all these years.

At this point, I would like to thank Professor Bjarni V. Haldòrsson, for inviting me to visit his group for a research internship. During this stay I enjoyed immensely his collaboration, his motivation, focus on his research and openness to collaborative work are to this day a great inspiration for me.

I would be remiss if I did not mention Professor Daniele Catanzaro, who introduced me to the topic of computational biology in the very beginning. But mostly, I would like to thank him for his support and friendship that have been invaluable to me and for being the person I've always looked up to.

Many thanks go out to all member of the Department of Computer Science that has offered me a contract as Teaching Assistant throughout my PhD. This experience made me passionate about teaching and taught me how to be a better teacher every day.

Then I would like my friends and colleagues both from G.O.M. and the Computer Science Department (former and current) for always being there to discuss science but also, and probably most importantly, for a coffee break or a small chat to recharge batteries. They provided a very pleasant working environment, which I believe played an important part in my achievements, and some of them are now very good friends.

And finally, I wish to convey special thanks to my family and my friends for helping and supporting me in so many ways throughout these years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

(alphabetical order)

ALU	*Arthrobacter Luteus*
ALUG	*ALU Graph*
B&B	*Branch-and-Bound*
B&C	*Branch-and-Cut*
B&P	*Branch-and-Price*
CRM	*Class Representative Model*
CG	*Column Generation*
CLHF	*Catanzaro-Labbé-Halldórsson ILP Formulation*
CLOH	*Consistent with a Loss of Heterozygosity*
COP	*Combinatorial Optimization Problem*
DNA	*Deoxyribonucleic Acid*
ELOH	*Evidence of a Loss of Heterozygosity*
GWAS	*Genome-Wide Association Studies*
IG	*Interval Graph*
ILOH	*Inconsistent whit a Loss of Heterozygosity*
ILP	*Integer Linear Programming*
IP	*Integer Programming*
LINE	*Long INterspersed Elements*

LOH *Loss of Heterozygosity*

LP *Linear Programming*

MCPP *Minimum Clique Partition Problem*

MILP *Mixed Integer Linear Programming*

MIPAIRP *Multiple Individuals Polymorphic ALU Insertions Recognition Problem*

MP *Master Problem*

MPPH *Minimum Perfect Phylogeny Haplotyping problem*

MPTG *Max-Point Tolerance Graph*

OR *Operations Research*

PAIRP *Polymorphic ALU Insertions Recognition Problem*

PPH *Pure Parsimony Haplotyping*

PPH-UD *Pure Parsimony Haplotyping under Uncertain Data*

PLOHP *Parsimonious Loss of Heterozygosity Problem*

RCI *Region Compatible with Insertions*

RMP *Reduced Master Problem*

RNA *Ribonucleic Acid*

SINE *Short INterspersed Elements*

SNP *Single Nucleotide Polymorphism*

1

INTRODUCTION

1.1 MOTIVATION

Model-based optimization is a key methodology in engineering employed in the design, analysis, construction, and operation of all kind of devices. Since engineering approaches are playing a growing role in the rapid evolution of systems biology [Wolkenhauer et al. 2005; Sontag 2005; Doyle et al. 2006; Kremling et al. 2007], it is reasonable to expect that mathematical optimization methods will contribute in a significant way to advances in systems biology. In fact, optimization is already playing a key role as given the several examples of applications of optimization in systems biology, offered in Greenberg et al. [2004], Błażewicz et al. [2005] and Banga [2008].

Optimization and optimality are certainly not new concepts in biology. The structures, movements, and behaviours of animals, and their life histories, have been shaped by the optimizing processes of evolution or of learning by trial and error [Alexander 1996; Sutherland 2005]. Moreover, optimization theory not only explains current adaptations of biological systems, but it also helps to predict new designs that may yet evolve [Alexander 1996; Sutherland 2005]. Some reviews on the use of optimization in the close fields of computational biology and bioinformatics can be found in Greenberg et al. [2004] and Larrañaga et al. [2006].

In the following section, we focus our attention on a specific methodology used in systems biology for the studies of common diseases and complex traits in human beings.

1.2 GENOME-WIDE ASSOCIATION STUDIES

The combinations of genetic and environmental factors are the causes of common diseases such as cancer, obesity, diabetes, cardiovascular, and inflammatory diseases [Consortium 2003]. Discovering these genetic factors would provide fundamental new insights into the diagnosis and treatment of human diseases. The recent completion of the sequencing phase of the Human Genome Project [Frazer et al. 2007] has shown that any two copies of the human genome differ from one another by approximately 0.1% of nucleotide sites i.e., one variant or polymorphism per 1000 nucleotides on average [Li et al. 1991; Wang et al. 1998; Cargill et al. 1999; Halushka et al. 1999]. This result has suggested as a possible approach to identifying genetic risk factors the search for an association between the variant sites of a specific chromosome region and a disease [Risch et al. 1996].

A number of association studies which focused on candidate genes, have already led to the discovery of genetic risk factors for several diseases. Examples include type 1 diabetes [Risch et al. 1996; Dorman et al. 1990; Nistico et al. 1996], type 2 diabetes [Deeb et al. 1998; Altshuler et al. 2000], Alzheimer's disease [Strittmatter et al. 1996], deep vein thrombosis [Dahlbäck 1997], inflammatory bowel disease [Hugot et al. 2001; Ogura et al. 2001; Rioux et al. 2001], hypertriglyceridemia [Pennacchio et al. 2001], schizophrenia [Stefansson et al. 2002], asthma [Van Eerdewegh et al. 2002], stroke [Gretarsdottir et al. 2003], and myocardial infarction [Ozaki et al. 2002].

One approach to perform association studies consists of testing each putative variant site for correlation with the disease (the direct or brute-force approach [Consortium 2003]). At present, this approach is limited to sequencing the functional parts of suspected genes, selected on the basis of a previous functional or genetic hypothesis [Consortium 2003]. Unfortunately, the direct approach entails the sequencing of numerous patient samples to identify the responsible variant sites and hence it is prohibitively expensive.

An alternative approach, called the indirect approach [Consortium 2003], consists of exploiting human sequence variation as genetic markers. In fact, over 90% of sequence variation among individuals is due to common variant sites [Li et al. 1991], most of which arose from single historical mutation events on the ancestral chromosome [Consortium 2005]. Hence, in a group of people affected by a disease, the variant sites causing the disease will be enriched in frequency compared with its frequency in a group of unaffected ones. This observation proved of considerable value, for example, in the identification of the genes responsible for cystic fibrosis and diastrophic dysplasia [Consortium 2003; Frazer et al. 2007].

The indirect approach is generally preferred to the direct one because it requires neither sequencing multiple patient samples nor prior knowledge of putative functional variant sites. However, in order to be applicable, the indirect approach requires determination of the common patterns of DNA sequence variation in the human genome , by characterizing sequence variants, their frequencies, and correlations between them [Consortium 2003]. In general, this is not an easy task because the current molecular sequencing methods only provide information about the combination (or conflation) of the paternal and the maternal of an individual (also called *genotype*) [Halldórs-

son et al. 2003].

Genome-Wide Association Studies (GWAS) is one of the most promising approach in the literature [Stranger et al. 2011], and has been able to successfully identify a very large number of polymorphism associated with disease (e.g. Styrkarsdottir et al. [2008], Gudbjartsson et al. [2008], and Rivadeneira et al. [2009]). It unfolds through the identification of a number of individuals carrying a disease or a trait and proceeds by comparing these individuals to those that do not carry or are not known to carry that disease/trait. Both sets of individuals are genotyped for a large number of genetic variants which are then tested for association with the disease/trait. Studies using tens of thousands of individuals are becoming common (or are widespreading) and are becoming increasingly the norm/standard in the study of association of genetic variants to disease [Styrkarsdottir et al. 2008; Gudbjartsson et al. 2008; Rivadeneira et al. 2009]. Since October 2010, 702 human GWASs have been published on 421 traits, the majority exhibiting medical relevance. "*The National Human Genome Research Institute at the National Institutes of Health*" weekly updates a catalogue of published GWAS results (`http://www.genome.gov/gwastudies`).

In this dissertation, our purpose is to illustrate the capabilities, opportunities and benefits that mathematical optimization can bring to research in systems biology.

First, we introduce several basic concepts to help the readers unfamiliar with mathematical optimization.

1.3 COMBINATORIAL OPTIMIZATION

The integer programming books by Wolsey [1998] and by Conforti et al. [2014] are the inspirational references on which the following

section is based.

The key elements of mathematical optimization problems are the *decision variables* (i.e. factors that can be varied during the search for the best solution), the *objective function* (i.e. the performance index which quantifies the quality of a solution defined by a set of decision variables, and which can be maximized or minimized), and the *constraints* (i.e. requirements that must be met, usually expressed as equalities and inequalities). Decision variables can be continuous (represented by real numbers), resulting in *continuous optimization* problems, or discrete (represented by integer numbers), resulting in integer optimization (also called *combinatorial optimization*) problems.

In many instances, there is a mix of continuous and integer decision variables resulting in a *Mixed Integer Linear Programming* (MILP) problem.

A *Mixed Integer Linear Programming* (MILP) is a problem which takes the following form:

$$\begin{aligned} \max_{x,y} \quad & c^T x + h^T y \\ \text{s.t.} \quad & Ax + Gy \leqslant b \\ & x \geqslant 0, \ \text{integral} \\ & y \geqslant 0 \end{aligned} \tag{P}$$

where the data are vectors $c^T = (c_1, \ldots, c_n)$ and $h^T = (h_1, \ldots, h_p)$, an $m \times n$ matrix $A = (a_{ij})$, an $m \times p$ matrix $G = (g_{ij})$ and a vector $b = (b_1, \ldots, b_m)^T$. The entries of c, h, A, G, b are considered as rational. The vectors $x = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_p)^T$ contain the decision variables of the problem. The elements of x are non-negative integers while the elements of y are non-negative real-valued numbers.

The set of feasible solutions to (P),

$$S := \{(x, y) \in \mathbb{Z}^n \times \mathbb{R}^p_+ : Ax + Gy \leqslant b\}$$

is called a *mixed-integer linear set*. While a *mixed 0,1 linear set* is set of the form

$$S := \{(x, y) \in \{0,1\}^n \times \mathbb{R}^p_+ : Ax + Gy \leqslant b\}$$

in which the integer variables are restricted to take value 0 or 1. A *mixed 0,1 linear program* is an mixed integer program whose set feasible solution is a mixed linear 0,1 set.

Solving a MILP is a hard problem, in general. A commonly used solution approach consists of (i) solving a relaxation (that is easier to solve numerically) and (ii) of giving a good approximation to the optimal objective value.

Given a mixed integer set $S \subseteq \mathbb{Z}^n \times \mathbb{R}^p$, a *linear relaxation* of S is a set of the form $P' := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : A'x + G'y \leqslant b'\}$ that contains S. A *Linear Programming* (LP) relaxation of (P) is a linear program $\max_{x,y}\{c^T x + h^T y : (x, y) \in P'\}$. For the mixed integer linear set S, there is a natural linear relaxation given by

$$P_0 := \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : Ax + Gy \leqslant b\}$$

obtained from S by discarding the integrality requirement on the vector x. The natural LP relaxation of (P) is, thus

$$\max_{x,y}\{c^T x + h^T y : (x, y) \in P_0\}$$

LP relaxations are used to provide bounds to the optimal solution because (i) it exists a very efficient algorithms to solve linear programs; (ii) one can generate a sequence of linear relaxations of S that provide increasingly tighter approximations of the set S.

The tightest formulation for the set S is given by its convex hull, which is the minimum convex set that contains S.

$$conv(S) := \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : (x, y) \text{ is a convex combination of points in } S\}$$

Figure 1 shows a mixed integer set S and two formulations P' and P'' for S. The convex hull of S is represented with a dashed line. The vector (c, h) indicates the direction in which one optimizes according

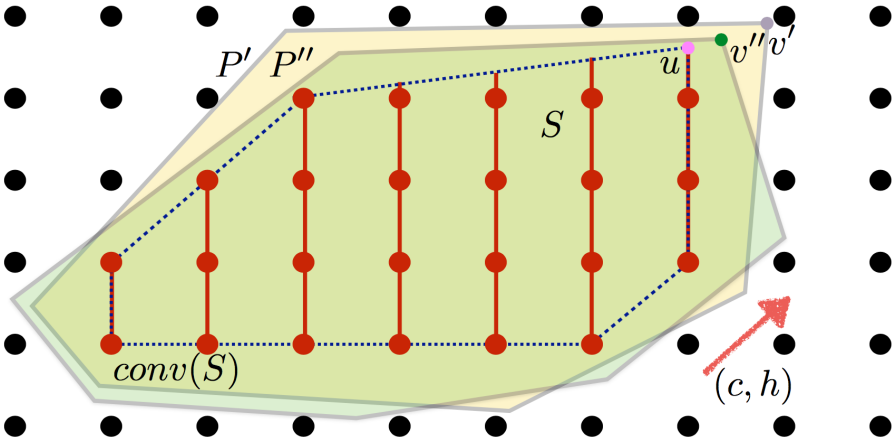


Figure 1: The polyhedral representation of a MILP problem

to the objective function. The point u is the optimum of the mixed integer problem one solves over the set S and the points v' and v'' are the solutions when one solves over the relaxations P' and P'', respectively. As such the value of the objective function at v' and v'' is an approximation of the optimum value, evaluated at u. One can see that P'' provides a better approximation than P' in the proximity of the optimum. The convex hull provides the best approximation to the mixed integer set S through linear inequalities. Unfortunately, for a given mixed integer set S, determining the linear inequalities that define the convex hull of the set can be a very difficult problem. Therefore, one strives to construct a model whose natural linear relaxation is a good approximation to the convex hull of feasible solutions and solves the natural LP relaxation of such a model to obtain a bound on the optimal solution. Once one has bounded on the solution, one of the following are standard solving schemes for a MILP can be used.

Branch-and-Bound

The *Branch-and-Bound* (B&B) method searches for an optimal solution to the MILP by branching, i.e. partitioning the set S into subsets, and attempting to prune the enumeration by bounding the subproblems

generated by the partition. In particular, at any point during the solution process, the status of the solution is described by a pool of yet unexplored subsets and the best solution found so far. Initially only one subset exists and the best solution found so far has an infinite objective value. The unexplored subspaces are represented as nodes in a dynamically generated search tree, which initially only contains the root, and each iteration of a classical B&B algorithm processes one such node. The iteration has three main components: (i) selection of the node to process, (ii) bound calculation,(iii) and branching.

The sequence of these may vary according to the strategy chosen for selecting the next node to process. If the selection of next subproblem is based on the bound value of the subproblems, then the first operation of an iteration after choosing the node is branching, i.e. subdivision of the solution space of the node into two or more subspaces to be investigated in a subsequent iteration. For each of these, it is checked whether the subspace contains an integer solution, in which case it is compared to the current best solution keeping the best of these. Otherwise the bounding function for the subspace is calculated and compared to the current best solution. If it can be established that the subspace cannot contain the optimal solution, the whole subspace is discarded, else it is stored in the pool of unexplored nodes together with it's bound.

This is called the eager strategy by Clausen et al. [1999] for node evaluation, since bounds are calculated as soon as nodes are available. The alternative is to start by calculating the bound of the selected node and then branch on the node if necessary. The nodes created are then stored together with the bound of the processed node. This strategy is called lazy by Clausen et al. [1999] and is often used when the next node to be processed is chosen to be a live node of maximal depth in the search tree.

The search terminates when there is no unexplored nodes left, and the optimal solution is then the one recorded as "current best".

Branch-and-Cut

In a B&B approach the tightness of the bound is crucial to prune the explored search tree. Tighter bounds can be obtained by applying cut generation at each of the subproblems in the search tree prior to branching. This leads to a *Branch-and-Cut* (B&C) approach. In some cases, the problem structure is such that the natural linear relaxation of (P) coincides with the convex hull of the mixed integer set S. In these cases, one refers to a perfect or ideal formulation. The main take-away from this section is the need to develop formulations for MILP problems, whose linear relaxations are as tight as possible, in the sense that they are good approximations for the convex hull of feasible solutions, at least in the vicinity of the optimum. The tighter a LP relaxation is for a MILP, the better the bound on the optimal solution. This can lead to exploring smaller search trees in the B&B solving procedure. To improve the tightness of the bound provided by the relaxation, one uses valid cuts to separate fractional points from the mixed integer set S.

Column Generation

Column generation is a general technique to solve linear programming problems with a very large number of variables. At each iteration, the method keeps a manageable subset of variables, solves the linear programming problem restricted to these variables, and either concludes that the optimal solution of the restricted problem corresponds to an optimal solution of the whole problem, or finds one or more "candidate variables" to improve the current solution.

Let us call the following linear program the *Master Problem* (MP).

$$\begin{aligned} z^\star_{MP} := \quad & \max_{x,y} \quad \sum_{j\in J} c^\top \lambda \\ & \text{s.t.} \quad \sum_{j\in J} \mathbf{a}_j \lambda \quad \leqslant \mathbf{b} \\ & \qquad\qquad \lambda_j \leqslant 0,\ j \in J \end{aligned} \tag{MP}$$

with $|J| = n$ variables and m constraints. In many applications n is exponential in m and working with the MP explicitly is not an option because of its sheer size. Instead, consider the *Reduced Master Problem* (RMP) which contains only a subset $J' \subseteq J$ of variables. An optimal solution $\boldsymbol{\lambda}^\star$ to the RMP needs not be optimal for the MP, of course. Denote by $\boldsymbol{\pi}^\star$ an optimal dual solution to the RMP. In the *pricing step* of the simplex method we look for a non-basic variable of negative reduced cost to enter the basis. To accomplish this in column generation, one solves the *pricing problem* (or *subproblem*).

$$z^\star_{PP} := \max\{c_j - \boldsymbol{\pi}^\star \mathbf{a}_j \mid j \in J\} \tag{PP}$$

When $z^\star_{PP} < 0$, the variable λ_j and its coefficient column $(c_j, \mathbf{a}_j)$ corresponding to a maximizer j are added to the RMP; this is solved to optimality to obtain optimal dual variable values, and the process iterates until no further improving variable is found. In this case, $\boldsymbol{\lambda}^\star$ optimally solves the MP as well. In particular, column generation inherits finiteness and correctness from the simplex method, when cycling is taken care of.

Branch-and-Price

To solve the MP, one can apply the B&B procedure, solving the MP relaxation by column generation to compute a bound at each node. This approach is known as *Branch-and-Price* (B&P). B&P is similar to B&P, except that procedure focuses on column generation rather than

row generation. In fact, *Pricing* and *Cutting* are complementary procedures for tightening a LP relaxation. In particular, B&P integrates B&B and column generation methods for solving large-scale *Integer Programming*s (IPs) . At each node of the B&B, columns may be generated to tighten (improve) the LP relaxation. In B&P, sets of columns are left out of the LP relaxation of large IPs because there are too many columns to handle efficiently and most of them will have their associated variables equal to zero in an optimal solution anyway. Then to check optimality, a sub-problem, also called the "*pricing problem*" is solved to identify columns to enter the basis. If such columns are found, the LP is re-optimized. Branching occurs when no columns "price" out to enter the basis and the LP solution does not satisfy integrality conditions.

1.4 OUTLINE OF THE THESIS

The thesis is organized as follows.

Chapter 2 studies an *NP*-hard problem called the *Pure Parsimony Haplotyping* (PPH). It is a combinatorial optimization problem which consists of finding the minimum number of haplotypes necessary to explain a given set of genotypes. PPH has attracted more and more attention in the recent years due to its importance in analyzing many fine-scale genetic data. Its application fields range from mapping of complex disease genes to inferring population histories, via through designing drugs, functional genomics and pharmacogenetics. Afterwards, we show our contribution in investigating, for the first time, a recent version of PPH called *Pure Parsimony Haplotyping under Uncertain Data* (PPH-UD). This version mainly arises when the input genotypes are not accurate, i.e., when some single nucleotide polymorphisms are missing or are affected by errors. We propose an exact ap-

proach to solution of PPH-UD based on an extend version of Catanzaro et al. [2010b] class representative model for PPH, currently the state-of-the-art integer programming model for PPH. We propose several reduction rules to improve the performances of the model and show how to embody the perfect phylogeny constrains in order to represent known familial relationships among members of the input population. The model is efficient, accurate, compact, polynomial-sized, easy to implement, solvable with any solver for mixed integer programming, and usable in all those cases for which the parsimony criterion is well suited for haplotype estimation.

Chapter 3 studies a second *NP*-hard combinatorial optimization problem occurring in computational biology called the *Parsimonious Loss of Heterozygosity Problem* (PLOHP). The problem consists of finding a *minimum cost clique cover* in a particular kind of interval graph called *Max-Point Tolerance Graph* (MPTG). The optimal solution to an instance of the PLOHP is of fundamental interest in GWAS as it allows for the association of major human diseases with chromosomic regions from patients that underwent to loss of heterozygosity events. Afterwards, we show our contribution by building on Catanzaro et al. [2013]' seminal work and present an ILP formulation for the PLOHP based on column generation. We introduce a number of preprocessing techniques to reduce the size of a given instance of the problem and we present decomposition strategies to divide a reduced instance into independent subproblems of even smaller size. We develop a new efficient algorithm to find maximum node-weighted cliques in MPTG. We embody this algorithm in a *Branch-and-Price* (B&P) algorithm for the PLOHP, whose computational performance is 10-30x faster than the previous approach described in the literature. The new algorithm is freely available and enables the solution of very large practical instances of the PLOHP, containing over 6 000 trios and SNPs.

Chapter 4 studies a third *NP*-hard combinatorial optimization problem arising in computational biology called the *Polymorphic ALU Insertions Recognition Problem* (PAIRP). The problem consists of partitioning suspected polymorphisms *Arthrobacter Luteus* (ALU) elements from a set of individuals into a minimum number of insertion areas. ALUs represent one of the most successful of all mobile elements, having a copy number well exceeding of 1 million copies in the human genome Lander et al. [2001] (contributing almost 11% of the human genome). Our contribution consists of extending Sveinbjörnsson et al. [2012]'s work on a single individual by providing a full combinatorial study of the version with multiple individuals called *Multiple Individuals Polymorphic ALU Insertions Recognition Problem* (MIPAIRP). Moreover, we show that the MIPAIRP can be formulated as a specific version of the clique partition problem in a particular class of graphs called *undirected unit interval graphs* and we prove its general *NP*-hardness. Our results give perspectives on the mathematics of the MIPAIRP and suggest new directions on the development of future efficient exact solution approaches. Finally, we provide a state-of-the-art integer programming formulation to exactly solve real instances of the MIPAIRP containing up to 3 000 individuals and 150 000 reads.

Finally, Chapter 5 summarize the main contributions of this thesis and outline directions for future research.

2

A CLASS REPRESENTATIVE MODEL FOR PURE PARSIMONY HAPLOTYPING UNDER UNCERTAIN DATA

Diploid organisms, such as humans, are characterized by having the DNA organized in pairs of chromosomes, of which 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 et al. 2001] showed that such copies are extremely similar and that the genomes of two different individuals are identical in more than 99% of the overall number of nucleotides. Nevertheless, differences at the genomic level (also known as polymorphisms) occur, on average, every 1,000 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 et al. 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) it is then called a *Single Nucleotide Polymorphism* (SNP). Specifically, a site is considered a SNP if for a minority 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 region define a *haplotype* (see Figure 2).

Haplotypes represent a fundamental source of information for disease association studies. In fact, over 90% of sequence variation among individuals is due to common variant sites, most of which arose from single historical mutation events on the ancestral chromosome [Li et al. 1991]. Hence, in a group of people affected by a disease, the SNPs causing or associated with the disease will be enriched in frequency compared with the corresponding frequencies in a group of unaffected individuals. This observation was of considerable assistance, for example, in the identification of the genes responsible for type 1 diabetes [Bell et al. 1984; Dorman et al. 1990; Nistico et al. 1996], type 2 diabetes [Altshuler et al. 2000; Deeb et al. 1998], Alzheimer's disease [Strittmatter et al. 1996], deep vein thrombosis [Dahlbäck 1997], inflammatory bowel disease [Hugot et al. 2001; Ogura et al. 2001; Rioux et al. 2001], hypertriglyceridaemia [Pennacchio et al. 2001], schizophrenia [Stefansson et al. 2002], asthma [Van Eerdewegh et al. 2002], stroke [Gretarsdottir et al. 2003], myocardial infarction [Ozaki et al. 2002], cystic fibrosis and diastrophic dysplasia [Consortium 2003, 2005].

Haplotyping an individual therefore consists of determining, for each copy of a given chromosome region, a pair of haplotypes. Haplotyping populations of individuals has attracted an increasing amount of attention in recent years [Helmuth 2001; Marshall 1999] because of its importance in the analysis of 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 to study an altered response of an organism to a particular therapy. In human pharmacogenetics, haplotypes explain why people react differently to different types or amounts of drugs. In fact, because SNPs affect the structure and function of proteins and enzymes, they may

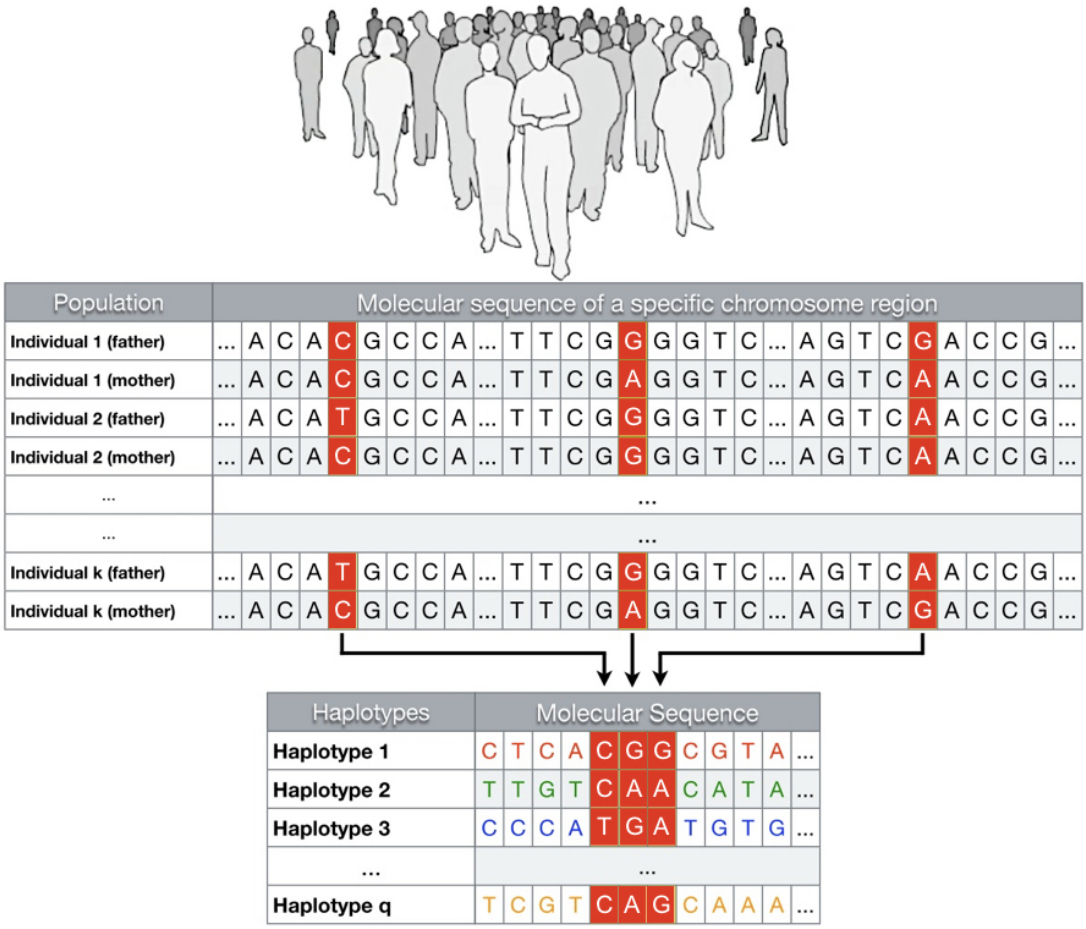


Figure 2: Any two copies of the human genome differ from one another by approximately 0.1% of nucleotide sites. In this example, most of the DNA sequence is identical in these chromosomes, but there are three nucleotides where variation occurs. A pattern of DNA sequence variation defines a haplotype.

influence the way in which a drug is absorbed and metabolized.

Direct sequencing of haplotypes via experimental methods is both time-consuming and expensive, and therefore current molecular sequencing methods generally provide more general genotype information. Specifically, genotype data provide information about the multiplicity of each SNP allele of a given individual, i.e., knowledge about its homozygous or heterozygous nature. Unfortunately, a drawback of using genotype data is that information regarding which heterozy-

gous site SNP variants came from the same chromosome copy remains unknown [Wang et al. 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], and Lancia et al. [2004] and can be summarized as follows: first, to experimentally obtain 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 et al. [1995], Fallin et al. [2000], Lancia et al. [2006], Niu et al. [2002a], Niu et al. [2002b], Stephens et al. [2003], and Stephens et al. [2001]).

In this chapter 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 et al. 2003]. Hence, because the number of distinct haplotypes observed in a population is much smaller than the number of possible haplotypes, the parsimony-based approaches aim to determine the minimum number of different haplotypes that, combined in pairs over time, have given rise to a set of observed genotypes.

The problem of haplotyping populations under parsimony (hereafter denoted as the *Pure Parsimony Haplotyping* (PPH) *problem*) is known to be *APX*-hard [Lancia et al. 2004]. This result justifies the development of several enumerative optimization algorithms that aim to solve exactly instances of PPH.

Specifically, Gusfield [2003] first proposed an integer-programming model to tackle instances of PPH. He described a model, exponential in size, characterized by two kinds of variables - one for haplotypes and the other one 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 used by Brown et al. [2004] and Bertolazzi et al. [2008]. To minimize the number of distinct haplotypes, Brown et al. [2004] proposed constructing haplotype vectors by associating a variable to each site; they subsequently used constraints to establish the exact haplotype structures. On the other hand, 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. Whereas Gusfield [2003] and Bertolazzi et al. [2008] used commercial mixed-integer programing solvers (CPLEX and Xpress-MP, respectively) to get solutions to their models, Brown et al. [2004] used a *Branch-and-Cut* (B&C) algorithm to solve their polynomial model. A comparison of their results shows that the Brown et al. [2004] polynomial model is well suited for big dimension samples, whereas the Gusfield [2003] and Bertolazzi et al. [2008] models are more efficient for medium dimensions, and specifically, when the recombination level (i.e., the parameter that affects the structure of the haplotypes) increases. So far, to our best knowledge, data sets containing 68 genotypes and 75 SNPs represent the limit size instances of PPH that can be exactly ana-

lyzed [Brown et al. 2006].

In the next section, we introduce some notations that will prove useful to state the PPH problem. In particular, we state the problem of detecting haplotypes from genotype data in terms of an optimization problem. Then, we extend the concepts to the general case in which SNPs are missing or affected by errors.

2.1 NOTATION AND PROBLEM STATEMENT

The human genome is divided in 23 pairs of chromosomes thereof, one copy is inherited from the father and the other from the mother. When a nucleotide site of a specific chromosome region shows a variability within a population of individuals then it is called SNP. Specifically, a site is considered a SNP if for a minority 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) [Aringhieri et al. 2011]. The least frequent allele, or *mutant type*, is generally encoded as '1', as opposed to the most frequent allele, or *wild type*, generally encoded as '0' [Zhang et al. 2006]. A *haplotype* is a set of alleles, or more formally, a string of length p over an alphabet $\Sigma = \{0, 1\}$ [Catanzaro et al. 2009].

Extracting haplotypes from a population of individuals is not an easy task. In fact, the current molecular sequencing techniques only provide information about the conflation of the paternal and maternal haplotypes of an individual (also called *genotype*) rather than haplotypes themselves [Halldórsson et al. 2003]. When the family-based genetic information of a population is available, haplotypes can be retrieved experimentally [Lu et al. 2003]. However, the experimental approach is generally laborious, cost-prohibitive, requires advanced

molecular isolation strategies [Clark et al. 2001], and sometimes not even possible [Lancia et al. 2004]. In absence of a family-based genetic information, a valid alternative to the experimental approach is provided by computational methods which estimate, by means of specific criteria, haplotypes from the set of genotypes extracted from a population of individuals.

Instance of PPH

Genotypes	SNP			
Genotype 1	1	1	1	1
Genotype 2	2	1	0	1
Genotype 3	0	1	2	1

Minimum Solution 1

Haplotypes	SNP			
Haplotype 1	1	1	0	0
Haplotype 2	0	0	1	1
Haplotype 3	1	0	0	1
Haplotype 4	0	1	1	0

Conflation				
Genotype 1	=	**Haplotype 1**	⊕	**Haplotype 2**
Genotype 2	=	**Haplotype 1**	⊕	**Haplotype 3**
Genotype 3	=	**Haplotype 2**	⊕	**Haplotype 4**

Minimum Solution 2

Haplotypes	SNP			
Haplotype 1	1	1	0	0
Haplotype 2	0	0	1	1
Haplotype 3	1	0	0	1
Haplotype 4	0	1	1	0

Conflation				
Genotype 1	=	**Haplotype 3**	⊕	**Haplotype 4**
Genotype 2	=	**Haplotype 1**	⊕	**Haplotype 3**
Genotype 3	=	**Haplotype 2**	⊕	**Haplotype 4**

Figure 3: Graphical representation of an instance of PPH and two alternative solutions

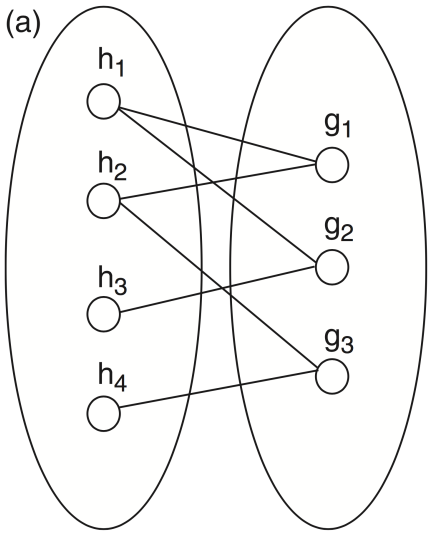
A genotype can be formally defined as a string of length p over an alphabet $\Sigma = \{0, 1, 2\}$, where the symbols '0' or '2' denote homozygous sites (of wild and mutant type, respectively) and the symbol '1' denotes heterozygous sites. As an example, the sequence $\langle 0, 2, 1 \rangle$ encodes a genotype in which: the first SNP is homozygous of wild type; the second SNP is homozygous of mutant type; and finally the third SNP is heterozygous. A genotype is said to be *degenerate* if it does not

contain '1's. A genotype g_k is said to be *resolved* from a pair of haplotypes $\{h_i, h_j\}$, in symbols $g_k = h_i \oplus h_j$, if the p-th entry of g_k, denoted as g_{kp}, is equal to the sum of the p-th entries of h_i and h_j, denoted as h_{ip} and h_{jp}, respectively. For example, the genotype $g_k = \langle 1, 2, 1, 0\rangle$ is resolved from $h_i = \langle 0, 1, 1, 0\rangle$ and $h_j = \langle 1, 1, 0, 0\rangle$. Haplotyping a set of genotypes $\mathcal{G}$ means finding the set of haplotypes resolving $\mathcal{G}$.

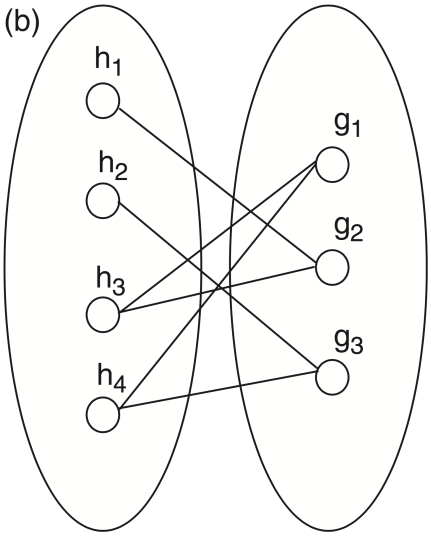
Any feasible solution to PPH is constituted by (i) a set $\mathcal{H}$ of (at most 2m) haplotypes and (ii) for each genotype $g_k \in \mathcal{G}$, a specification of a pair of haplotypes, say $\{h_i, h_j\}$, resolving g_k, i.e., such that $g_k = h_i \oplus h_j$. For example, the alternative (minimum) solutions relative to the instance of PPH shown in Figure 3 satisfy both conditions (i) and (ii). However, note that, although having the same set of haplotypes, Solutions 1 and 2 of Figure 3 are different, as they satisfy condition (ii) in a different way.

Hence, a feasible solution to PPH can be represented by means of a bipartite graph in which each vertex $g_k \in \mathcal{G}$ is of degree 2 and the two other vertices, say h_i and h_j, adjacent to g_k satisfy $g_k = h_i \oplus h_j$. As an example, the bipartite graphs corresponding to Solutions 1 and 2 of Figure 3 are depicted in Figure 4a and 4b, respectively. The bipartite graph representation of a solution suggests that in a feasible solution to PPH the haplotypes induce a family of subsets of genotypes satisfying the following three properties: (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 haplotypes of Solution 1 in Figure 3 induce the family of subsets of Figure 5a satisfying properties (i)-(iii). Specifically, the subsets are induced by the following four haplotypes: $h_1 = \langle 1, 1, 0, 0\rangle$ inducing the subset $S_1 = \{\langle 1, 1, 1, 1\rangle, \langle 2, 1, 0, 1\rangle\}$, $h_2 = \langle 0, 0, 1, 1\rangle$ inducing the subset $S_2 = \{\langle 1, 1, 1, 1\rangle, \langle 0, 1, 2, 1\rangle\}$, $h3 = \langle 1, 0, 0, 1\rangle$ inducing the subset $S_3 = \{\langle 2, 1, 0, 1\rangle\}$, and finally $h_4 = \langle 0, 1, 1, 0\rangle$ inducing the subset $S_4 = \{\langle 0, 1, 2, 1\rangle\}$. Similarly, the haplotypes of Solution 2 in Figure 4

induce the family of subsets of Figure 5b satisfying properties (i)-(iii). Specifically, the subsets are induced by the following four haplotypes: $h_1 = \langle 1,1,0,0\rangle$ inducing the subset $S_1 = \{\langle 2,1,0,1\rangle\}$, $h_2 = \langle 0,0,1,1\rangle$ inducing the subset $S_2 = \{\langle 0,1,2,1\rangle\}$, $h_3 = \langle 1,0,0,1\rangle$ inducing the subset $S_3 = \{\langle 1,1,1,1\rangle, \langle 2,1,0,1\rangle\}$, and finally $h_4 = \langle 0,1,1,0\rangle$ inducing the subset $S_4 = \{\langle 1,1,1,1\rangle, \langle 0,1,2,1\rangle\}$. It is worth noting that,



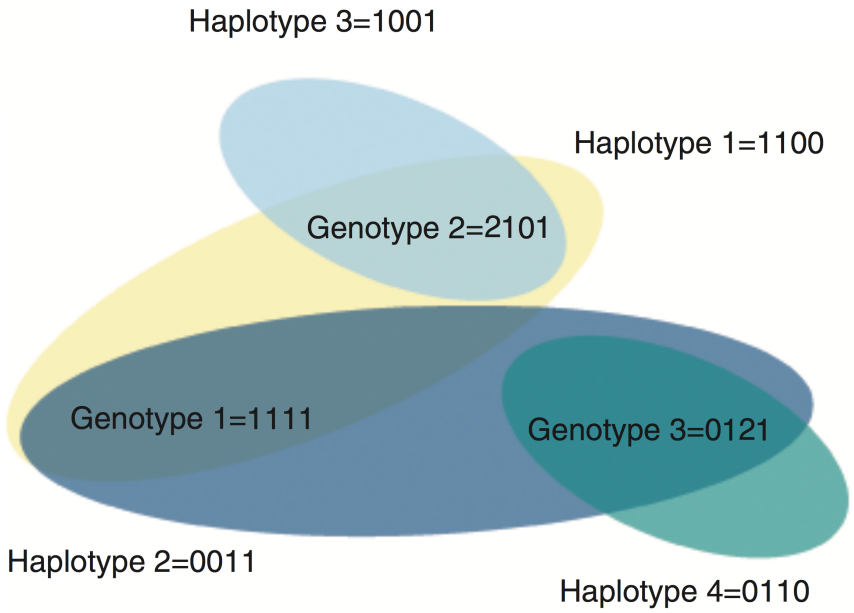
(a) Solution 1



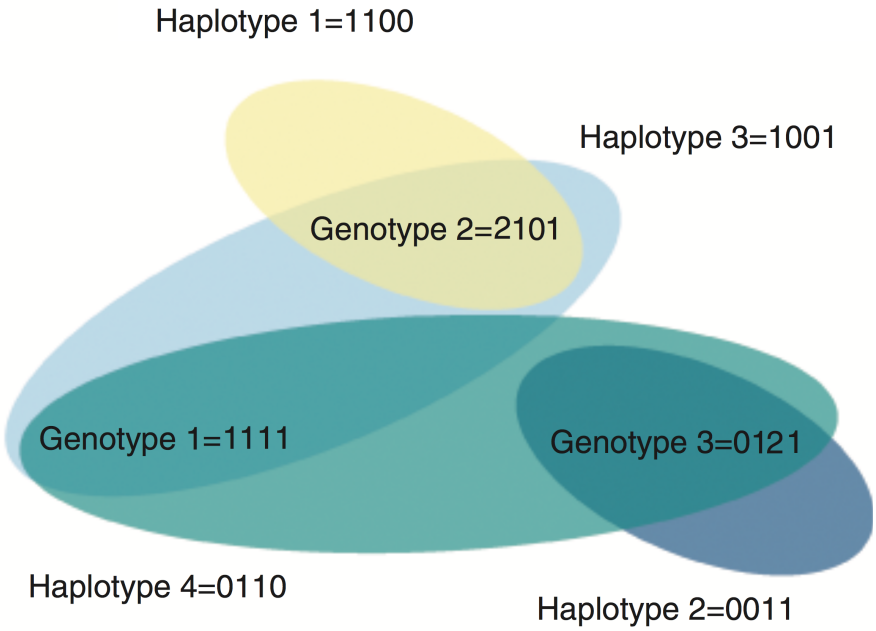
(b) Solution2

Figure 4: Bipartite graph representation of Solutions 1 and 2 of Figure 3.

given a genotype and denoted n as the number of its heterozygous



(a) Solution 1



(b) Solution 2

Figure 5: Examples of subsets of genotypes induced by the two alternative solutions of Figure 3.

sites, there exist 2^{n-1} possible haplotypes that may resolve it [Lancia et al. 2004]. This insight entails the use of a criterion to select pairs of

haplotypes among plausible alternatives. Gusfield [2001] and Wang et al. [2003] observed that the number of distinct haplotypes existing in a large population of individuals is generally much smaller than the overall number of distinct genotypes observed in that population. This insight has suggested that, for low-rate recombination genes at least, the criterion of minimizing the overall number of haplotypes necessary to resolve a set of genotypes may have good chances to recover the biological haplotype set. This criterion, formally introduced by Gusfield [2001], is known as *the pure parsimony criterion of haplotype estimation* and was of considerable assistance, for example, in the identification of the genes responsible for psoriasis and severe alopecia areata [Aringhieri et al. 2011]. Haplotyping a set of genotypes under the parsimony criterion involves solving an optimization problem, called *pure parsimonious haplotyping problem*, that can be stated as follows:

Problem 1 (*The Pure Parsimony Haplotyping (PPH) problem*)**.** *Given a set* $\mathcal{G}$ *of* m *non-degenerating genotypes, having* s *SNPs each, find the minimum set* $\mathcal{H}$ *of haplotypes such that for each genotype* $g_k \in \mathcal{G}$ *there exists a pair of haplotypes* $\{h_i, h_j\} \in \mathcal{H}$ *resolving* g_k.

PPH is known to be polynomially solvable when each genotype has at most two heterozygous sites [Lancia et al. 2006], and *NP*-hard when each genotype has at least three heterozygous sites [Lancia et al. 2004]. Recently, Brown et al. [2006] introduced an interesting version of PPH called the *Pure Parsimony Haplotyping under Uncertain Data* (PPH-UD). This version mainly arises when the input genotype set $\mathcal{G}$ is not accurate, i.e., when some SNPs are missing or affected by errors, a situation that often occurs in practice. In this case, the input of the problem may include also a binary matrix B, called *the error mask matrix*, whose generic entry b_{kp} is equal to 1 if the p-th SNP of genotype g_k is uncertain (i.e., missing or affected by an error), and 0 otherwise. When a

given SNP is uncertain its actual value could sensibly deviate from its true value. For example, the true value of a wild type homozygous SNP affected by uncertainty could be homozygous of mutant type or even heterozygous. Similarly, the true value of a heterozygous SNP affected by uncertainty could be homozygous of wild or mutant type. The presence of uncertain data modifies the standard definition of resolution for a genotype. Specifically, Brown et al. [2006] stated that when uncertainty occurs in the input data a genotype g_k is resolved by a pair of haplotypes $\{h_i, h_j\}$ if $g_{kp} = h_{ip} + h_{jp} + b_{kp}e_{kp}$, for all SNPs p, being e_{kp} being an integer variable taking values in the set $\mathcal{E} = \{-2, -1, 0, 1, 2\}$. Brown et al. [2006] described an integer programming model able to tackle instances of PPH affected by uncertain data. Unfortunately, the authors did not offer experimental evidence of the performances of their model due to its unpractical runtimes.

In the next section we address this critical issue by introducing a possible integer linear programming model to solve exactly instances of PPH-UD. The model is based on an extension of Catanzaro et al. [2010b] *Class Representative Model* (CRM), currently one of the best integer programming model for PPH described in the literature. We propose several reduction rules to improve the performances of our model and show how to embody perfect phylogeny constrains in order to represent known familial relationships among members of the input population.

2.2 DESIGN AND IMPLEMENTATION

As shown in Catanzaro et al. [2010b], any feasible solution of PPH induces a family of subsets of genotype such that: (i) each subset represents one unique haplotype with elements in the subset being

genotypes carrying the haplotype, (ii) each genotype belongs to exactly two subsets, and (iii) every pair of subsets intersects in at most one genotype. This principle can be exploited also when dealing with PPH-UD. Specifically, let associate an index to each subset S of genotypes induced by a haplotype h. If i is the smallest index of a genotype belonging to S, then i is the index associated to S and the subset will be denoted as S_i. Since each genotype k belongs to exactly two subsets (as it must be explained by exactly two haplotypes) 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 the index set $Q = K \cup K'$, where $K = \{1, \ldots, m\}$ and $K' = \{1', \ldots, m'\}$. Assume that an order is defined on Q in such a way that $1 < 1' < 2 < 2' < \ldots < m < m'$. Define x_i, $\forall\, i \in Q$, 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, $\forall\, k \in K, \forall\, i, j \in Q$, as a decision variable equal to 1 if genotype k belongs to the subsets S_i and S_j, and 0 otherwise. Denote $\mathcal{P}$ as the set of the input SNPs and z_{ip}, $\forall\, i \in Q,\ p \in \mathcal{P}$, 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} describe explicitly the haplotypes of the solution.

For every non-null entry of the error mask matrix B denote e_{kp}^c as a decision variable accounting for the difference between the value

of g_{kp} and the true underlying value. Specifically, e^c_{kp} is equal to 1 if the p-th entry of genotype g_k is corrected with a value $c \in C = \{-2, -1, 1, 2\}$, and 0 otherwise. Finally, let E^{LB} and E^{UB} be a lower and an upper bounds on the overall number of errors in $\mathcal{G}$. Then, the following model is a valid formulation of PPH-UD:

Formulation 1. *Class Representative Model (CRM) for PPH-UD*

$$\min \sum_{i \in Q} x_i \tag{1.1}$$

$$\text{s.t.} \quad x_{i'} \leqslant x_i \qquad \forall i \in Q \tag{1.2}$$

$$\sum_{i,j \in Q} y^k_{ij} \geqslant 1 \qquad \forall k \in K \tag{1.3}$$

$$\sum_{j \in Q} y^k_{ij} \leqslant x_i \qquad \forall k \in K,\ \forall i \in Q \tag{1.4}$$

$$y^k_{kk'} \leqslant x_{k'} \qquad \forall k \in K \tag{1.5}$$

$$z_{kp} + z_{k'p} = g_{kp} \qquad \forall k \in K, \forall p \in \mathcal{P} : b_{kp} = 0 \tag{1.6}$$

$$z_{kp} + z_{k'p} = e^1_{k,p} + 2e^2_{k,p} \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 0, b_{kp} = 1 \tag{1.7}$$

$$z_{kp} + z_{k'p} = 2 + (-e^{-1}_{k,p} - 2e^{-2}_{k,p}) \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 2, b_{kp} = 1 \tag{1.8}$$

$$z_{kp} + z_{k'p} = 1 + (e^1_{k,p} - e^{-1}_{k,p}) \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 1, b_{kp} = 1 \tag{1.9}$$

$$z_{ip} \leqslant 1 - \sum_{j \in Q} y^k_{ij} \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 0,\ b_{kp} = 0,$$
$$\forall i \in Q : i \neq k, k' \tag{1.10}$$

$$z_{ip} \geqslant \sum_{j \in Q} y^k_{ij} \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 2,\ b_{kp} = 0,$$
$$\forall i \in Q : i \neq k, k' \tag{1.11}$$

$$z_{ip} + z_{jp} \geqslant y^k_{ij} \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 1,\ b_{kp} = 0, \forall i, j \in Q \tag{1.12}$$

$$z_{ip} + z_{jp} \leqslant 2 - y_{ij}^{k} \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 1,\ b_{kp} = 0,\ \forall i, j \in Q \tag{1.13}$$

$$z_{ip} \leqslant 1 - \sum_{j \in Q} y_{ij}^{k} + (e_{k,p}^{1} + 2e_{k,p}^{2}) \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 0,\ b_{kp} = 1,\ \forall i \in Q, i \neq k, k' \tag{1.14}$$

$$z_{ip} \geqslant \sum_{j \in Q} y_{ij}^{k} + (-e_{k,p}^{-1} - 2e_{k,p}^{-2}) \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 2,\ b_{kp} = 1,\ \forall i \in Q, i \neq k, k' \tag{1.15}$$

$$z_{ip} + z_{jp} \geqslant y_{ij}^{k} + (e_{k,p}^{1} - e_{k,p}^{-1}) \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 1,\ b_{kp} = 1,\ \forall i, j \in Q \tag{1.16}$$

$$z_{ip} + z_{jp} \leqslant 2 - y_{ij}^{k} + (e_{k,p}^{1} - e_{k,p}^{-1}) \qquad \forall k \in K,\ \forall p \in \mathcal{P} : g_{kp} = 1,\ b_{kp} = 1,\ \forall i, j \in Q \tag{1.17}$$

$$e_{k,p}^{1} + e_{k,p}^{2} \leqslant 1 \qquad \forall k \in K, \forall p \in \mathcal{P} : g_{kp} = 0,\ b_{kp} = 1 \tag{1.18}$$

$$e_{k,p}^{-1} + e_{k,p}^{-2} \leqslant 1 \qquad \forall k \in K, \forall p \in \mathcal{P} : g_{kp} = 2,\ b_{kp} = 1 \tag{1.19}$$

$$e_{k,p}^{1} + e_{k,p}^{-1} \leqslant 1 \qquad \forall k \in K, \forall p \in \mathcal{P} : g_{kp} = 1,\ b_{kp} = 1 \tag{1.20}$$

$$\sum_{k \in K} \sum_{p \in \mathcal{P} : b_{kp} = 1} \sum_{c \in C} e_{kp}^{c} \geqslant E^{LB} \tag{1.21}$$

$$\sum_{k \in K} \sum_{p \in \mathcal{P} : b_{kp} = 1} \sum_{c \in C} e_{kp}^{c} \leqslant E^{UB} \tag{1.22}$$

$$x_i, z_{ip}, y_{ij}^{k}, e_{kp}^{c} \in \{0, 1\} \tag{1.23}$$

The objective function (1.1) 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 (1.2) implies that if the haplotype h_i is not used, then $h_{i'}$ should not be used. Constraints (1.3) impose that each genotype g_k must belong to exactly two subsets S_i, S_j, and constraints (1.4) 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 (1.5) are a consequence of the definition of the dummy genotype k'. Actually, they constitute a special version of constraints (1.4) when genotype k is resolved by haplotype k'. Constraints (1.6) impose the sum operation among haplotypes in absence of uncertainty. Constraints (1.7-1.9) translate the sum operation among haplotypes when uncertainty occurs in the input data. Specifically, constraints (1.7) account for the

correction imposed on the p-th SNP of haplotypes h_i and $h_{i'}$ when g_i has its p-th SNP equal to 0. In this case, two situations may occur at the p-th SNP: either no correction is performed, or a correction is performed by setting to 1 one between e^1_{kp} and e^2_{kp}. If a correction is performed and e^1_{kp} is set to 1 at the p-th SNP then one haplotype will be homozygous of wild type and the other of mutant type. On the contrary, if e^2_{kp} is set to 1 then both haplotypes will be homozygous of mutant type. Constraints (1.8) account for the correction imposed on the p-th SNP of haplotypes h_i and $h_{i'}$ when g_i has its p-th SNP equal to 2. Similarly to constraints (1.7), also in this case two situations may occur: either no correction is performed, or a correction is performed by setting to 1 one between e^{-1}_{kp} and e^{-2}_{kp}. If a correction is performed and e^{-1}_{kp} is set to 1 at the p-th SNP then one haplotype will homozygous of wild type and the other of mutant type. On the contrary, if e^{-2}_{kp} is set to 1 then both haplotypes will be homozygous of wild type. Finally, constraints (1.9) account for the correction imposed on the p-th SNP of haplotypes h_i and $h_{i'}$ when g_i has its p-th SNP equal to 1. If a correction is performed and e^1_{kp} is set to 1 at the p-th SNP then both haplotypes will be homozygous of mutant type. On the contrary, if e^{-1}_{kp} is set to 1 then both haplotypes are homozygous of wild type. Constraints (1.10) establish the relations between variables z_{is} and y^k_{ij} in absence of uncertainty. Specifically, they force the p-th site 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 (1.11) force the p-th site of the haplotype h_i to be equal to 1 when at least one genotype g_k, whose p-th entry equal to 2, belongs to the induced subset S_i. Constraints (1.12-1.13) force one of the two p-th sites of haplotypes h_i and h_j to be equal to 1 when the p-th entry of genotype g_k is equal to 1. Constraints (1.14-1.17) are the analogous version of constraints (1.10-1.13) in presence of uncertainty in the input data. Constraints (1.18-1.20) impose that at most one variable e^c_{kp} can be equal to one in presence of uncertainty in

the input data. Finally, constraints (1.21-1.22) impose the upper and lower bounds on the error variables e^c_{kp}.

2.2.1 *Reducing model size*

The particular nature of the set of indices Q can be exploited to reduce the size of the CRM for PPH-UD. As observed in Catanzaro et al. [2010b], this operation proves fundamental to improve the efficiency of whole model.

Given that $y^k_{ij} = 1$ 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^k_{ij} only when $i < j \leqslant k$ or $i = k$ and $j = k'$. For example, variable $y^2_{1,1'}$ does not need to be defined as well as all variables $y^k_{ii'}$ for all $i, k \in K$, $k \neq i$. Similarly, variables $y^k_{ik'}$ or $y^k_{k'i}$ (depending on whether k is smaller or bigger than i) do not need to be defined for $i \in Q$ with $i \neq k$ and $i \neq k'$. In fact, if $y^k_{ik'} = 1$, 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^k_{ij} are redundant and assuming that variable $y^1_{11'} = 1$, it is easy to see that variables y^k_{ij} do not need to be defined whenever they belongs to one of the following sets:

$$R_1 = \{y^k_{ij} : k \in K,\ i, j \in K \cup K', j < i < k\}, \tag{2}$$

$$R_2 = \{y^k_{ik'} : k \in K,\ i \in K \cup K', i \leqslant (k-1)'\}, \tag{3}$$

$$R_3 = \{y^k_{ii'} : k \in K,\ i \in K \cup K', 2 \leqslant i \leqslant k-1\}. \tag{4}$$

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 $g_{ip} = 0$, $g_{jp} = 0$, $g_{kp} = 1$, and $b_{ip} = b_{jp} = b_{kp} = 0$,

variable y_{ij}^{k} is necessarily equal to 0 since the containment of 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 that violate the haplotype sum operator, we see that the following proposition holds:

Proposition 1. *The set of variables*

$$R_4 = \{y_{ij}^{k} : i, j, k \in K, p \in \mathcal{P}, g_{kp} = 1, \quad (5)$$
$$g_{ip} = g_{jp} \neq 1, b_{kp} = b_{ip} = b_{jp} = 0\}$$

is redundant.

Proof. By contradiction, if the set R_4 is not redundant, then for some $p \in \mathcal{P}$, genotype k may belong to two subsets S_i and S_j. Let assume $g_{kp} = 1$ and $g_{ip} = g_{jp} = 0$ and that uncertainty does not affects the p-th SNP. Since haplotype h_i is associated with the subset S_i and explains genotype i, it must have the p-th SNP equal to 0 otherwise the sum operator would be violated. At the same time, since haplotype h_j is associated with the subset S_j and explains genotype j, it must have the p-th SNP equal to 0 otherwise the sum operator would be violated. In turn, this implies that genotype k cannot be resolved by h_i and h_i since a necessary condition for its resolution is that the p-th SNP of h_i or h_j be equal to 1. In the second case, let assume $g_{kp} = 1$ and $g_{ip} = g_{jp} = 2$ and that uncertainty does not affects the p-th SNP. Since haplotype h_i is associated with the subset S_i and explains genotype i, it must have the p-th SNP equal to 1 otherwise the sum operator would be violated. At the same time, since haplotype h_j is associated with the subset S_j and explains genotype j, 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 and h_i since a necessary condition for its resolution is that the p-th SNP of h_i or h_j be equal to 0.Hence, $y_{ij}^{k} = 0$ in any feasible solution of PPH. □

Similar arguments can be used to prove the following proposition:

Proposition 2. *The sets of variables*

$$R_5 = \{y_{ij}^k : i,j,k \in K, p \in \mathcal{P}, g_{kp} = 0, \quad (6)$$
$$g_{ip} = 2 \text{ or } g_{jp} = 2, b_{kp} = b_{ip} = b_{jp} = 0\}$$
$$R_6 = \{y_{ij}^k : i,j,k \in K, p \in \mathcal{P}, g_{kp} = 2, \quad (7)$$
$$g_{ip} = 0 \text{ or } g_{jp} = 0, b_{kp} = b_{ip} = b_{jp} = 0\}$$

are 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 $g_{ip} = 2$ and that uncertainty does not affects the p-th SNP. Since haplotype h_i is associated with the subset S_i and explains genotype i, 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, independent of j, $y_{ij}^k = 0$ in any feasible solution of PPH. A similar approach can be used to prove that R_6 is redundant. □

Note that, removing the redundant variables y_{ij}^k of Propositions 1 and 2 can be performed in $O(m^3 s)$. Finally, a similar process of reduction can be applied to variables z_{ip} both by removing those whose value is fixed by constraints (1.6) (e.g., when $g_{kp} = 0$ or when $g_{kp} = 2$). In this way, only variables z_{ip} involved in constraints (1.6) when $g_{kp} = 1$ and in constraints (1.7-1.9) need to be defined.

2.2.2 *Accounting for perfect phylogeny constraints*

In some practical situations it may turn out useful to know not only the haplotype set that resolves a given set of genotypes, but also how

the familial relationships among members of a population of individuals reflect on the estimated haplotypes. This version of PPH is known in the literature as *the Minimum Perfect Phylogeny Haplotyping problem* (MPPH) *problem* and was first introduced by Bafna et al. [2004]. MPPH requires to find a minimum set of haplotypes resolving the input genotypes and forming a *perfect phylogeny*, i.e., a haplotype set $\mathcal{H}$ that does not contain the following matrix [Brown et al. 2006; Pe'er et al. 2004]:

$$F = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Excluding matrix F from the haplotype set $\mathcal{H}$ is equivalent to impose the following constraints on z_{ip} variables:

$$\alpha_1 z_{ip_1} + \alpha_2 z_{ip_2} + \alpha_3 z_{jp_1} + \alpha_4 z_{jp_2} + \alpha_5 z_{kp_1} + \alpha_6 z_{kp_2} + \alpha_7 z_{qp_1} + \alpha_8 z_{qp_2} \leqslant 3 \quad \forall\, i \neq j \neq k \neq q \in Q,\ p_1 \neq p_2 \in \mathcal{P} \tag{8}$$

$$\sum_{s=1}^{8} \alpha_s = 0 \tag{9}$$

$$\alpha_s \in \{-1, 1\} \tag{10}$$

Adding (8), (9) and (10) to the class representative model could drastically slow down any mixed integer linear programming solver due to the large number of constraints to be added. For this reason, these constraints should be imposed dynamically (e.g., by means of a B&C algorithm). Note also that not all instances of PPH-UD admit a perfect phylogeny. For example, if the genotype set $\mathcal{G}$ contains a sub-matrix

$$\Omega = \begin{pmatrix} 0 & 0 \\ 0 & 2 \\ 2 & 0 \\ 2 & 2 \end{pmatrix}$$

whose entries are not affected by uncertainty, then we can conclude that $\mathcal{G}$ does not admit solution. The same situation persists if any row of Ω (possibly all) contains a heterozygous site. Thus, when a perfect phylogeny is required, preprocessing techniques based on sufficient conditions (such as those described in Bafna et al. [2003] able to exclude a-priori the existence of a solution satisfying a perfect phylogeny could be used to speed up the computation. Testing the performances of the CRM for PPH-UD under the perfect phylogeny requirement is out of the scope of the present thesis and warrants additional analysis.

2.3 EXPERIMENTAL RESULTS

In this section we analyze the performances of the class representative model to solve instances of the pure parsimony haplotyping problem under uncertain data. Similar to Brown et al. [2006] and Catanzaro et al. [2010b], we 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 the accuracy of our algorithm to haplotype inference solvers that do not use the parsimony criterion. This analysis has been already performed by Gusfield [2003], Wang et al. [2003], and Marchini et al. [2006], and we refer the interested reader to their respective articles.

2.3.1 *Implementation*

The results have been obtained by implementing the CRM for PPH-UD in Mosel 2.0 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). In our experiments we activated the Xpress-MP Optimizer automatic cuts, the Xpress-MP presolving strategy, and used the Xpress-MP primal heuristic to generate the first upper bound.

2.3.2 *Benchmark instances*

As in Catanzaro et al. [2010b], we used the standard Brown et al. [2006]'s datasets for testing the performances of our model. Specifically, through Hudson's MS program [Hudson 2002], Brown and Harrower 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. Both the uniform and non-uniform datasets consist of collections of 30 or 50 genotypes having 10, 30, 50, 75 or 100 SNPs each. Each dataset contains a number of instances variable between 15 and 50. The authors also considered biological data from chromosomes 10 and 21, over all four Hap-Map [Consortium 2003] populations. For each input the authors selected sequences having 30, 50, and 75 SNPs, respectively, giving a total of 8 datasets consisting of 3 instances each. Brown and Harrower's

datasets are not subjected to uncertainty, for this reason we considered a three sets of random generated error mask matrices having an error ratio (i.e., the number of entries equal to 1 divided by mp) equal to 5%, 10%, and 15% respectively. Brown et al. [2006]'s datasets and the error mask matrices used in our experiments can be downloaded at the address: http://homepages.ulb.ac.be/~lporrett/PPHUD/PPHerr.zip.

In Tables 1-3 we show the performances of the CRM for PPH-UD under different error ratios. Specifically, the columns of Tables 1-3 evidence the average, 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), and the number of nodes expanded in each group of instances belonging to a given dataset.

2.3.3 *Computational performances*

In order to obtain a qualitative measure of the running time performances of the CRM for PPH-UD, we compared the numerical results of the model with the corresponding ones of the CRM for PPH (RM version, see Catanzaro et al. [2010b] running on the same datasets in absence of uncertainty. The performances of the CRM for PPH are shown in Table 4. Moreover, in order to obtain a qualitative measure of the accuracy of the CRM for PPH-UD, we compared the optimal solutions provided by CRM for PPH in absence of uncertainty with the corresponding ones provided by CRM for PPH-UD. Specifically, fixed a generic instance of PPH-UD, we computed the accuracy of the CRM for PPH as the ratio between the number of matching haplotypes in the solutions provided by both the CRM for PPH-UD and the CRM for PPH divided the overall number of haplotypes in the solution provided by

CRM for PPH. The accuracy of the CRM for PPH-UD under increasing error ratios it is shown in Table 5. For sake of notation, in the following subsections we shall denote CRM1 and CRM2 as the CRM for PPH and the CRM for PPH-UD, respectively.

2.3.3.1 *Uniform Datasets*

The numerical experiments relatives to the uniform datasets showed that CRM2 takes sensibly more time than CRM1 to solve Brown et al. [2006]'s datasets, confirming the hardness of PPH-UD with respect to PPH. Specifically, while CRM1 took in average 8 seconds to solve the most difficult dataset having 10 SNPs, CRM2 took in average 10 seconds independently from the error ratio, and even longer on instances 03, 05, 06, 08 and 11 of dataset 50x10r4 where 19.657, 62.160, 34.429, 40.416, and 15.974 seconds were needed to find the optimum. This trend persists also in the instances having 30 SNPs, in which CRM1 took in average 11.772 seconds while CRM2 needed an average solution time of 43.273 seconds when considering an error ratio of 5%, with the exception of instances 02, 08, 09, 11, 13 and 14 which needed 62.182, 51.462, 60.079, 60.020, 122.443, and 58.514 seconds, respectively. We experienced also a generalized decrease of the average solution time when considering an error ratio of 10% and, vice versa, an increase of the average solution time when considering an error ratio of 15%. When considering instances having a larger number of SNPs, we experienced a generalized increment of the average solution time taken by CRM2, proportional to the increment of the error ratio. Interestingly, the average gap and number of branches performed by CRM2, although not directly comparable with the corresponding one of CRM1, results relatively small, confirming the tightness of the class representative model also for uncertain data.

Dataset	Time (sec.)			Gap (%)			Nodes		
	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform									
50x10	9.775	35.814	2.030	0.000	0.000	0.000	1.000	1	1
50x10r4	8.699	62.160	2.642	0.000	0.000	0.000	1.000	1	1
50x10r16	11.055	44.405	2.635	0.000	0.000	0.000	2.067	7	1
50x30	43.273	122.443	7.878	1.338	5.882	0.000	5.533	49	1
30x50	36.903	178.408	4.270	1.620	9.091	0.000	34.760	827	1
30x75	40.422	115.258	10.835	1.383	6.250	0.000	9.900	37	1
30x100	98.348	354.685	7.330	0.484	2.590	0.000	43.700	244	1
Non-Uniform									
50x10	12.878	135.199	1.576	1.000	8.333	0.000	1.000	1	1
50x30	42.623	193.406	3.422	0.498	4.762	0.000	14.667	111	1
30x50	32.124	120.886	1.576	0.837	4.762	0.000	62.467	562	1
30x75	110.105	323.692	29.245	0.644	3.819	0.000	203.200	1317	3
30x100	639.736	7210.800	45.658	0.642	4.000	0.000	2527.867	34214	6
Biological									
CHR10-CEU	25.568	76.186	0.033	5.000	5.000	0.000	83.667	249	1
CHR21-CEU	12.137	32.109	0.593	0.000	0.000	0.000	16.333	27	1
CHR10-HBC	54.896	150.916	1.570	2.381	7.143	0.000	121.000	321	1
CHR21-HBC	33.719	100.581	0.071	10.317	16.667	0.000	9.667	23	1
CHR10-JPT	5.204	14.500	0.003	1.668	5.000	0.000	5.000	13	1
CHR21-JPT	9.691	19.591	0.021	1.830	5.490	0.000	43.667	123	1
CHR10-YRI	2864.727	7254.880	49.311	0.889	2.667	0.000	5464.000	13956	77
CHR21-YRI	2551.640	7651.670	0.165	8.684	26.051	0.000	128.000	382	1

Table 1: Performances of the CRM for PPH-UD when considering input data having an error ratio of 5%.

Dataset	Time (sec.)			Gap (%)			Nodes		
	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform									
50x10	9.641	47.717	2.461	0.000	0.000	0.000	1.000	1	1
50x10r4	16.057	62.160	2.663	0.000	0.000	0.000	1.067	2	1
50x10r16	13.838	44.178	2.678	0.000	0.000	0.000	2.667	9	1
50x30	38.338	85.872	9.874	1.445	8.889	0.000	6.733	35	1
30x50	39.799	222.460	3.249	1.192	8.333	0.000	40.020	1113	1
30x75	43.441	138.641	10.366	1.579	6.250	0.000	22.800	58	1
30x100	120.666	323.663	17.943	0.303	2.778	0.000	78.800	538	1
Non-Uniform									
50x10	13.174	126.922	1.765	8.333	0.000	0.000	1.000	1	0
50x30	40.194	84.860	2.418	0.919	5.882	0.000	23.933	265	1
30x50	27.455	73.425	1.488	0.922	4.737	0.000	18.467	66	1
30x75	108.737	325.864	33.529	0.814	4.348	0.000	250.733	1539	3
30x100	1563.970	7208.110	37.634	0.773	4.000	0.000	11673.800	74593	2
Biological									
CHR10-CEU	32.592	95.712	0.209	0.000	0.000	0.000	144.667	431	1
CHR21-CEU	4.935	11.259	0.544	1.852	5.556	0.000	2.000	4	1
CHR10-HBC	185.619	529.879	2.228	2.381	7.143	0.015	1739.333	4926	1
CHR21-HBC	42.578	127.162	0.074	10.317	16.667	0.000	33.000	93	1
CHR10-JPT	19.037	56.795	0.003	1.667	5.000	0.000	33.667	99	1
CHR21-JPT	8.433	19.635	2.716	2.225	6.667	0.000	33.667	99	1
CHR10-YRI	2992.214	7231.800	30.751	0.877	2.632	0.000	6538.000	14597	9
CHR21-YRI	2534.816	7600.780	1.083	4.119	12.356	0.000	117.000	349	1

Table 2: Performances of the CRM for PPH-UD when considering input data having an error ratio of 10%.

Dataset	Time (sec.)			Gap (%)			Nodes		
	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform									
50x10	10.092	57.006	1.560	0.000	0.000	0.000	1.000	1	1
50x10r4	11.570	33.895	3.275	0.000	0.000	0.000	1.000	1	1
50x10r16	9.209	18.627	2.790	1.068	8.333	0.000	1.467	5	1
50x30	44.276	157.222	7.467	1.123	5.882	0.000	4.133	15	1
30x50	39.772	222.039	6.187	1.006	8.333	0.000	42.700	1149	1
30x75	49.558	160.645	8.048	1.409	6.250	0.000	19.500	67	1
30x100	97.879	262.453	23.986	0.664	3.836	0.000	32.800	98	1
Non-Uniform									
50x10	9.925	126.922	2.113	1.000	8.333	0.000	1.133	1	1
50x30	38.689	84.860	4.000	0.400	5.882	0.000	11.267	265	1
30x50	33.272	73.425	1.793	0.614	4.737	0.000	54.400	66	1
30x75	88.030	325.864	32.260	0.800	4.348	0.000	95.400	1539	3
30x100	631.495	7207.900	50.928	1.157	4.270	0.000	2371.200	32622	12
Biological									
CHR10-CEU	27.340	68.299	1.669	0.000	0.000	0.000	53.667	155	3
CHR21-CEU	13.455	36.601	0.522	0.000	0.000	0.000	6.667	12	1
CHR10-HBC	87.770	248.825	2.228	4.347	7.143	0.015	384.333	1127	1
CHR21-HBC	39.878	118.866	0.075	10.317	16.667	0.000	27.000	69	1
CHR10-JPT	23.436	69.457	0.002	1.667	5.000	0.000	72.667	213	1
CHR21-JPT	2403.091	7188.610	2.781	2.222	6.667	0.000	4406.333	13125	1
CHR10-YRI	702.533	1777.450	62.414	0.000	0.000	0.000	583.000	1635	48
CHR21-YRI	2545.198	7630.880	1.500	5.449	16.346	0.000	123.333	368	1

Table 3: Performances of the CRM for PPH-UD when considering input data having an error ratio of 15%.

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
30x50	8.922	47.467	0.73	1.694	7.69231	0	10.260	75	1
30x75	15.624	35.693	1.358	1.649	6.66667	0	24.300	92	1
30x100	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
30x50	10.764	24.108	0.815	0.890	4.09091	0	20.533	61	1
30x75	22.389	61.869	3.537	1.038	5.55556	0	62.286	387	1
30x100	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
CHR21-CEU	18.868	54.562	0.428	1.515	4.54545	0	49.667	145	1
CHR10-HBC	38.058	96.324	8.746	2.593	7.77778	0	67.000	151	1
CHR21-HBC	0.182	0.456	0.017	0.000	0	0	8.000	19	1
CHR10-JPT	0.895	1.583	0.368	1.515	4.54545	0	7.000	11	1
CHR21-JPT	1.781	2.87	0.967	0.833	2.5	0	15.667	29	1
CHR10-YRI	73.723	116.127	31.353	1.111	3.33333	0	89.667	123	63
CHR21-YRI	2349.331	6819.2	50.012	0.000	0	0	3815.667	11199	123

Table 4: Performances of the CRM for PPH (RM version) on Brown et al. [2006]'s datasets.

Dataset	5(%)			10(%)			15(%)		
	Average	Max	Min	Average	Max	Min	Average	Max	Min
Uniform									
50x10	99.01	100.00	92.31	100.00	100.00	100.00	98.02	100.00	90.91
50x10r4	99.51	100.00	88.89	99.02	100.00	88.89	99.51	100.00	88.89
50x10r16	98.22	100.00	84.62	98.22	100.00	92.31	96.00	100.00	75.00
50x30	96.85	100.00	78.57	92.91	100.00	70.59	94.88	100.00	71.43
30x50	90.22	100.00	57.14	89.83	100.00	50.00	84.24	100.00	25.00
50x50	94.17	100.00	85.00	91.75	100.00	70.59	87.38	100.00	56.25
30x75	88.07	100.00	62.50	86.36	100.00	62.50	77.84	100.00	6.25
30x100	94.89	100.00	82.35	90.91	100.00	75.00	90.34	100.00	64.71
Non-Uniform									
50x10	92.61	100.00	78.57	96.55	100.00	86.67	94.58	100.00	81.25
30x50	86.87	100.00	68.42	85.52	100.00	68.75	81.14	100.00	59.09
50x50	88.65	100.00	80.00	87.60	100.00	69.57	85.22	100.00	62.07
30x75	80.97	100.00	56.52	79.46	100.00	66.67	80.97	100.00	56.52
30x100	82.18	95.00	72.00	75.00	95.71	65.00	78.74	95.71	60.00
Biological									
CHR10-CEU	83.33	100.00	70.83	78.79	86.36	70.83	68.18	80.00	54.17
CHR21-CEU	91.67	100.00	83.33	58.33	75.00	33.33	56.25	83.33	27.78
CHR10-HBC	68.29	92.86	23.53	53.66	85.71	23.53	51.22	80.00	17.65
CHR21-HBC	99.81	100.00	68.42	65.63	100.00	47.37	71.88	100.00	52.63
CHR10-JPT	71.43	100.00	55.00	66.67	90.91	45.00	59.52	100.00	30.00
CHR21-JPT	69.81	80.95	60.00	50.94	57.14	41.18	43.40	52.94	33.33
CHR10-YRI	66.28	80.00	52.78	53.49	84.00	30.56	48.84	68.00	25.00
CHR21-YRI	50.63	100.00	35.85	41.77	100.00	20.75	40.51	100.00	26.42

Table 5: Accuracy of the CRM for PPH-UD under different error ratios.

In terms of accuracy, the performances of CRM2 in the uniform datasets are very good. Specifically, the average accuracy is over 90% in the majority of the analyzed datasets and independently from the error ratio. However, it is worth noting that in some instances the accuracy may decrease sensibly (see, e.g., datasets 30x50 and 30x75) and proportionally to the increment of the error ratio, by suggesting, as general trend, the fact that the higher the error ratio the more difficult is to recover the correct haplotype set.

2.3.3.2 *Nonuniform Datasets*

The general trend observed in the uniform datasets persists also in the nonuniform datasets. Specifically, as shown in Table 2, CRM2 took in average 10 times more than the average solution time taken by CRM1 to solve instances having 10 SNPs, reaching a maximum solution time of 135.199 seconds when tackling instance 06 affected by an error ratio of 5%. Similarly, when tackling instances having 30 SNPs, CRM2 took in average 4 times more than the average solution time taken by CRM1, reaching a maximum solution time of 193.406 seconds when tackling instance 00 affected by an error ratio of 5%. When dealing with instances having more than 30 SNPs, CRM2 took sensibly more than CRM1 reaching a solution time of 7210.800 seconds when tackling the instance 100-30.03 affected by an error ratio of 5%.

In terms of accuracy, the performances of CRM2 in the nonuniform datasets are still good, but slightly poorer than in the uniform datasets. Specifically, the average accuracy is over 80% in the majority of the analyzed datasets and independently from the error ratio. Similarly to the uniform datasets, in some instances the accuracy may decrease sensibly (see, e.g., datasets 30x75). However, in the worst case, the decrease turns out to be much smaller than the corresponding one in the uniform datasets.

2.3.3.3 *Biological Datasets*

To complete the performance analysis on Brown et al. [2006]'s datasets, we tested CRM2 on the biological datasets. Once again, the general trend observed in the uniform and nonuniform datasets persists also in the biological datasets: CRM2 is sensibly slower than CRM1, a part from datasets CHR10-CEU and CHR21-CEU in which the trend is inverted due to the peculiar nature of both datasets. While the average gap of CRM1 never overcomes 2.6%, the average gap of CRM2 overcomes 10% confirming the hardness of the biological datasets. However, we stress once again the fact that PPH and PPH-UD are de facto two different problems, hence intrinsic values such as the gap cannot be directly compared. Our analysis just aims at offering experimental evidence of the tightness of the class representative model in tackling instances of the pure parsimony haplotyping problem under uncertain data.

The small number of instances constituting each biological dataset (three instances per dataset) prevents a clear statistical characterization of the performances of CRM2 in terms of accuracy. As general trend, we have observed that the accuracy approaches 100% in the majority of the biological datasets analyzed. Nevertheless, in a number of datasets this trend changes, leading the accuracy level to low values. Investigating the reason why this phenomenon arises and the possible corresponding remedies warrants additional analysis.

In the next Chapter we introduce another problem arising when the genotypes of a child and the ones of his two parents are known and a deletion polymorphism may be observed as an event on the child chromosome, which is one of the main factors giving rise to several human diseases.

3

A BRANCH-AND-PRICE ALGORITHM FOR THE PARSIMONIOUS LOSS OF HETEROZYGOSITY PROBLEM

Single Nucleotide Polymorphisms (SNPs), together with the recombination process, constitute the predominant form of human variation and are as such source of human disease [Li et al. 1991; Wang et al. 1998; Cargill et al. 1999; Halushka et al. 1999; Terwilliger et al. 1998; Hoehe et al. 2000; Catanzaro et al. 2010a]. A large number of other types of variations exist in nature, including insertions, inversions, translocations. One type of variation being *deletions*, which occur when a subsequence of the human genome is present in a reference genome but is not in the genome of an individual being analyzed [Catanzaro et al. 2013]. When the genotypes of a child and its two parents are known, a deletion polymorphism may be observed as a *Loss of Heterozygosity* (LOH) event on the child chromosome. In fact, the laws of Mendelian inheritance state that each individual inherits one copy of a chromosome from the father and one from the mother. Hence, for a given SNP, an individual can be either homozygous, i. e., the nucleotides of the parental DNA strands are equal, or heterozygous, i. e., the nucleotides of the parental DNA strands are different. For example, the first individual in Figure 6 is homozygous at the first SNP and heterozygous at the second SNP. When a deletion polymorphism occurs, an individual carries only a single copy of the chromosomal segment while the other is missing. As an example, the first individual in Figure 6 carries a deletion at the third SNP of the considered chromosome region (denoted by the symbol '-'). A deletion is *de novo* if it is only found in the child and not in one of its parents. Otherwise,

Population	Molecular Sequence of a specific chromosome region																												
Father (1st DNA strand)	...	A	C	A	C	G	G	T	C	...	T	T	C	G	G	G	G	T	C	...	A	G	T	C	G	A	C	C	G	...
Father (2st DNA strand)	...	A	C	A	C	G	G	T	C	...	T	T	C	G	A	G	G	T	C	...	A	G	T	C	—	A	C	C	G	...
Mother (1st DNA strand)	...	A	C	A	T	G	G	T	C	...	T	T	C	G	G	G	G	T	C	...	A	G	T	C	A	A	C	C	G	...
Mother (2st DNA strand)	...	A	C	A	C	G	G	T	C	...	T	T	C	G	G	G	G	T	C	...	A	G	T	C	A	A	C	C	G	...
Child (1st DNA strand)	...	A	C	A	T	G	G	T	C	...	T	T	C	G	G	G	G	T	C	...	A	G	T	C	A	A	C	C	G	...
Child (2st DNA strand)	...	A	C	A	C	G	G	T	C	...	T	T	C	G	A	G	G	T	C	...	A	G	T	C	G	A	C	C	G	...

Figure 6: In this example, we compare the DNA sequences from a set of individuals correlated between each other (father, mother and child). These SNPs are located in the three variant sites. The symbol '-' represents a deletion, i. e., a lack of a nucleotide

the deletion is said to be *inherited* i. e., passed from one of the two parents to the child. If the deletion event modifies the heterozygosity of an individual at a given site of a chromosomal region then we say that a LOH event occurred at that site [Catanzaro et al. 2013].

Deletions may have noxious effects on an individual as they may give rise to several human diseases. For example, recent studies showed that schizophrenia [Stefansson et al. 2008], multiple sclerosis [Halldórsson et al. 2011a], Alzheimer [Goedert et al. 2006], type I diabetes [Elder et al. 2001], obesity [Shinawi et al. 2011], and some cardiovascular diseases [Momma et al. 1999; Ogilvie et al. 2009; Puvabanditsin et al. 2010] are associated with large recurrent deletion events occurring across the genomes of affected individuals [McClellan et al. 2010]. Detecting deletions across the genome of individuals could be of fundamental assistance for the diagnosis and the treatment of certain human diseases, hence increasing research efforts have been dedicated to this task in recent years [Frazer et al. 2007].

A natural approach to perform the task consists of comparing the genomes of a given population of affected individuals with the genomes from a population of unaffected ones. However, the genomes of the individuals are generally not readily available. Moreover, even if they were, the comparison process would be laborious, time consuming

and cost-prohibitive due to the large amount of data to analyze [Catanzaro et al. 2009]. Hence, computational methods may constitute a valid alternative to the experimental approach. In this context, a number of methods have been suggested for the detection of deletions, including tiling arrays [Conrad et al. 2010] and high throughput sequencing [Korbel et al. 2009; Chen et al. 2009]. Similarly to Catanzaro et al. [2013] and Halldórsson et al. [2011a], in this chapter we focus on detecting germline deletions from genotype data of an offspring and his parents. These data may be derived from SNP arrays, which have been used for genome-wide association studies at a number of research labs(see e.g., Halldórsson et al. [2011a], Conrad et al. [2006], Speed et al. [2007] and McCarroll et al. [2008]).

Halldórsson et al. [2011a] observed that detecting deletions from genotype data may not be straightforward due to the limit of both genotyping technology and the presence of uncertainty in the genotyping process. In fact, SNP genotyping technology is not able to discern easily the difference between a homozygous site and a deletion, hence the output will always be a homozygous SNP even if the true genotype of the individual may carry only a single copy of the genotype. Moreover, even if a deletion polymorphism were observed in molecular data, such an event could be due either to the presence of real deletions or to genotyping errors, i. e., misreadings caused by the genotyping technology. Catanzaro et al. [2013] proposed a possible approach to address these major limitations. In particular, the authors introduced the *parsimony criterion* (see Albert [2005]) to identify chromosomic regions from a population of individuals that underwent to a massive loss of heterozygosity events. Subsequently, they stated such a criterion in terms of a combinatorial optimization problem, called the *Parsimonious Loss of Heterozygosity Problem* (PLOHP), consisting of finding a minimum clique covering on a particular class of interval graphs called *Max-Point Tolerance Graph* (MPTG) (see Section

3.1). The authors proved the *NP*-hardness of the PLOHP, investigated some fundamental properties of MPTGs, and proposed an *Integer Linear Programming* (ILP) formulation able to optimally solve instances of the PLOHP containing up to 2000 individuals and 2000 SNPs within 5 hours of computing time. Unfortunately, the proposed formulation proved unable to optimally solve instances of the PLOHP involving a larger number of individuals, a desirable requirement in disease association studies.

Starting from Catanzaro et al. [2013]' seminal work, in this chapter we present an ILP formulation for the PLOHP based on column generation. We introduce a number of preprocessing techniques to reduce the size of a given instance of the problem and we investigate possible decomposition strategies to divide a reduced instance into a family of independent subproblems having even smaller size. We develop a new efficient algorithm to find maximum node-weighted cliques in MPTGs and we embody this algorithm in a *Branch-and-Price* (B&P) algorithm for the PLOHP, that turned out to be 10-30x faster than the previous approach described in the literature. The new algorithm is freely available and enables the solution of very large practical instances of the PLOHP, containing over 6000 trios and SNPs, within 1 hour of computing time. Before describing the solution approach, in the next section we introduce some useful notation and definitions.

3.1 NOTATION AND PROBLEM FORMULATION

Consider a *trio* t, i. e., a set of two parents and an offspring (see Figure 7), and denote s as a given SNP genotyped in t. Then, one of the following three situations may occur:

1. The SNP s can be *Inconsistent whit a Loss of Heterozygosity* (ILOH), a situation that occurs when the child is heterozygous. In this

Individuals	SNPs				
Father (1st DNA strand)	...	A	A	C	...
Father (2st DNA strand)	...	A	A	C	...
Mother (1st DNA strand)	...	G	A	A	...
Mother (2st DNA strand)	...	G	C	A	...
Child (1st DNA strand)	...	A	A	C	...
Child (2st DNA strand)	...	G	A	C	...

Figure 7: An example of a *trio*, i. e., a set of two parents and their offspring, and their genotypes. SNPs are highlighted in red. Specifically, the first highlighted column represents a SNP inconsistent with a loss of heterozygosity; the second highlighted column represents a SNP consistent with a loss of heterozygosity; the third highlighted column represents a SNP showing an evidence of a loss of heterozygosity.

case the two alleles must have been inherited from each parent. For example, this is the case for the first highlighted column of the sequences of the trio shown in Figure 7.

2. The SNP s can be *Consistent with a Loss of Heterozygosity* (CLOH), a situation that occurs when a deletion may (but needs not) be introduced to explain the trio's inheritance pattern. Confer the second highlighted column of the sequences of the trio shown in Figure 7, the SNP of the child could be explained by means of a deletion of the paternal pattern.

3. The SNP s can show *Evidence of a Loss of Heterozygosity* (ELOH), a situation that occurs when a deletion or a genotyping error are the only possible explanation for the trio inheritance pattern. Confer the third highlighted column of the sequences of the trio shown in Figure 7, the SNP of the child can be explained only by means of a deletion of the maternal pattern.

According to the above definitions, a trio genotyped at m SNPs can be encoded as a string of length m over an alphabet $\Sigma = \{1, 0, X\}$, where '1' codes for a SNP inconsistent with having a loss of heterozy-

gosity; 'o' codes for a SNP consistent with a loss of heterozygosity; and 'X' codes for a SNP showing evidence of loss of heterozygosity [Halldórsson et al. 2011a]. For example, the string $t = \langle X100X0X010 \rangle$ represents a trio genotyped at 10 SNPs, thereof 5 consistent with having a loss of heterozygosity, 3 showing evidence of a loss of heterozygosity, and 2 inconsistent with having a loss of heterozygosity. As shown in Catanzaro et al. [2013], not all of the ELOH-SNPs in a trio are relevant for disease association studies. In particular, the ones that are relevant are only those that are in common with other trios. The identification of the relevant ELOH-SNPs in a set $\mathcal{T}$ of n trios genotyped at m SNPs can be performed by solving a *NP*-hard combinatorial optimization problem that can be stated as follows.

Consider a trio $t \in \mathcal{T}$ and let p be an ELOH-SNP in t. Since a trio t can also be seen as a string of length m over Σ, with a little abuse of notation we also denote p as the position of the p-th character in t. For example, given the trio $t = \langle X100X0X010 \rangle$, $p = 5$ denotes both the second ELOH-SNP in t and the fifth character in the string t. Let $I_p^t = \{(l_p, r_p), x_p\}$ be a *point-interval* for a ELOH-SNP $p \in t$, i. e., a pair constituted by an interval $(l_p, r_p) : 0 < l_p < r_p < m + 1$ and a point $x_p \in (l_p, r_p)$. Let us associate a point-interval I_p^t to each ELOH-SNP in t, in such a way that x_p denotes an ELOH-SNP located at the p-th character of t and r_p (respectively l_p) represents the position of the closest ILOH-SNP on the right (respectively left) of x_p. Whenever an ambiguity may arise, we shall write x_p^t to denote that the point x_p is associated to the point-interval I_p^t. If such an ILOH-SNP on the right (respectively left) is missing, we set $r_p = m + 1$ (respectively $l_p = 0$). For example, given the trio $t = \langle X100X0X010 \rangle$ we can associate to the first ELOH-SNP the point-interval $I_1^t = \{(0, 2), 1\}$; to the second ELOH-SNP the point-interval $I_5^t = \{(2, 9), 5\}$; and to the third ELOH-SNP the point-interval $I_7^t = \{(2, 9), 7\}$.

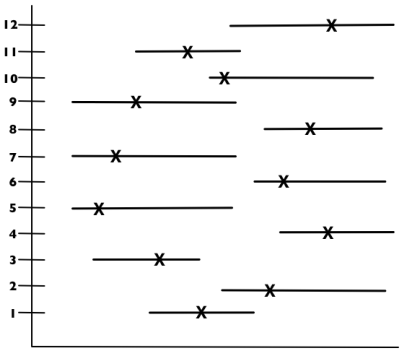


Figure 8: An example of point-intervals

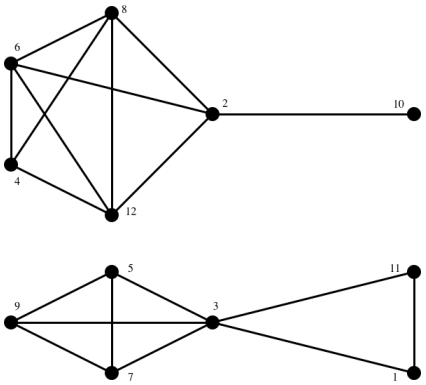


Figure 9: The MPTG induced by the point-intervals shown in Figure 8.

Given a set $\mathcal{T}$ of n trios genotyped at m SNPs, denote $\mathcal{I}$ as the set of point-intervals associated to the ELOH-SNPs of all of the trios $t \in \mathcal{T}$ and $I_k = \{(l_k, r_k), x_k\}$ as the k-th point-interval in $\mathcal{I}$.

Let $G = (V, E)$ be the *induced graph of* $\mathcal{T}$, i. e., a graph having a vertex $v \in V$ for each point-interval $I_v \in \mathcal{I}$ and an edge between two distinct vertices u and v iff $x_u, x_v \in (l_u, r_u) \cap (l_v, r_v)$. Then, the problem of identifying the relevant ELOH-SNPs in $\mathcal{T}$ that can be associated with a given disease can be stated in terms of the following combinatorial optimization problem [Catanzaro et al. 2013]:

Problem 2. *Parsimonious Loss of Heterozygosity Problem (PLOHP). Given a set $\mathcal{T}$ of n trios genotyped at m SNPs, the induced graph $G = (V, E)$ and a cost function f that associates a positive value to a maximal clique C of G, find a cover $\mathcal{C}^*$ of G into singletons and maximal cliques such that*

$$\mathcal{C}^* = \arg \min_{C \subseteq \mathcal{C}(G)} \sum_{C \in \mathcal{C}^*} f(C)$$

where $\mathcal{C}(G)$ is the set of all maximal cliques in G and singletons are cliques of cardinality 1.

The graph G is a *Max-Point Tolerance Graph* (MPTG) [Catanzaro et al. 2014], i. e., it is a generalization of *Interval Graphs* (IGs) (namely, intersection graphs of intervals on a line [Booth et al. 1976]) in which

each pair of intervals can "tolerate" a non-empty intersection (without forming an edge) as long as both of their distinguished points are not contained in this intersection. For example, Figure 9 shows the max-point-tolerance graph induced by the point-intervals shown in Figure 8. Combinatorial optimization problems that are *NP*-hard in general often turn out to be solvable in polynomial time on IGs (see e.g., Fishburn [1985] and Fulkerson et al. [1965]). However, this fact does not hold in general on MPTGs (see Catanzaro et al. [2014]). In particular, the PLOHP has been shown to be *NP*-hard [Catanzaro et al. 2013].

In this specific application, the cost function f is such that $f(S) = \psi(|S|)$, for any $S \subseteq V$ and for some $\psi : [0 \dots |V|] \to \mathbb{R}_0^+$. In particular, both Catanzaro et al. [2013] and Halldórsson et al. [2011a] defined f as

$$f(C) = \begin{cases} c_1 & \text{if } |C| = 1 \\ c_2 & \text{if } |C| \geqslant 2, \end{cases} \qquad \forall\, C \in \mathcal{C}(G) \tag{11}$$

where c_1 and c_2 are two constants such that $0 < c_1 \leqslant c_2$. Generally, there are different costs depending on the SNP and deletion should be considered. In fact, genotyping technologies are usually characterized by a high variability in the quality of the SNP genotypes produced Consortium 2003. A common method for dealing with these is to remove from analysis markers that show many ELOH events Conrad et al. 2006, this method however may remove most of the signal from the data in the preprocessing step. Similarly, different regions in the genome may have different propensity for carrying deletions Korbel et al. 2009. Based on Catanzaro et al. [2013] analysis about different values of c_1 and c_2 pursuit the assumption made by Halldórsson et al. [2011a] in which functions $f(c)$ always assign the same cost to each ELOH and error, respectively. Remark that the approach developed here can be extended to any general linear cost set function.

The *NP*-hardness of the PLOHP justifies the development of solution approaches such as those described in the next section.

3.2 A BRANCH-AND-PRICE ALGORITHM FOR THE PLOHP

In this section we describe an exact solution algorithm for the PLOHP based on *Branch-and-Price* (B&P). Throughout the thesis we assume, without loss of generality, that the cost function f in the PLOHP is a *size-defined submodular set function*, i.e., it is such that $f(S) = \psi(|S|)$, for any subset $S \subseteq V$ and for some $\psi : [0 \dots |V|] \to \mathbb{R}_0^+$. This characterization of f is commonly assumed in loss of heterozigosity studies [Catanzaro et al. 2013] and allows to solve the PLOHP in polynomial time when the MPTG is reducible to an interval-graph [Gijswijt et al. 2007].

3.2.1 *An integer linear programming formulation for the PLOHP*

Consider a set $\mathcal{I}$ of point-intervals and let $G = (V, E)$ be the MPTG induced by $\mathcal{I}$. Then, the following proposition holds [Catanzaro et al. 2013]:

Proposition 3. *A MPTG contains at most* $|\mathcal{I}|(|\mathcal{I}| - 1)/2$ *maximal cliques.*

As the number of maximal cliques in a MPTG is polynomially bounded and a polynomial time algorithm to determine all of them exists (see Catanzaro et al. [2013]), a polynomial sized formulation for the PLOHP can be obtained as follows. Consider a vertex $v \in V$ and let $\mathcal{C}_v = \{C \in C(G) : v \in C\}$. Let x_v be a decision variable equal to 1 if vertex $v \in V$ forms a singleton in a solution to the problem and 0 otherwise. Similarly, let y_C be a decision variable equal to 1 if the maximal clique $C \in \mathcal{C}(G)$, with $|C| \geqslant 2$, is selected in a solution to the problem and 0 otherwise. Then, a valid ILP formulation for the PLOHP is:

Formulation 2. *Catanzaro-Labbé-Halldórsson ILP Formulation (CLHF)*

$$z_{BIP} =$$

$$\min \sum_{\substack{C \in \mathcal{C}(G) \\ |C| \geqslant 2}} f(C) y_C + \sum_{v \in V} f(v) x_v \tag{12a}$$

$$\text{s.t.} \sum_{\substack{C \in \mathcal{C}_v(G) \\ |C| \geqslant 2}} y_C + x_v \geqslant 1 \qquad \forall\, v \in V \tag{12b}$$

$$x_v \in \{0,1\} \qquad \forall\, v \in V \tag{12c}$$

$$y_C \in \{0,1\} \qquad \forall\, C \in \mathcal{C}(G), |C| \geqslant 2. \tag{12d}$$

The objective function (12a) minimizes the cost of finding a clique cover of G, and constraints (12b) impose that, for each vertex $v \in V$, the solution contains either a clique or a singleton in $C(G)$ that covers v. Note that, as the overall number of maximal cliques in a MPTG is polynomially bounded, CLHF is polynomial-sized. However, when the instances of the PLOHP are very large (e.g., they include more than 250,000 point-intervals), the use of CLHF may become unpractical due to the large number of y variables. A possible approach to overcome such a limitation consists of reformulating CLHF by inverse projection [Martin 1999]. To this end, denote $\mathcal{C}^\tau(G)$ as a subset of $\mathcal{C}(G)$ that includes $\tau < |\mathcal{C}(G)|$ maximal cliques in G and set $\mathcal{C}_v^\tau(G) = \{C \in \mathcal{C}^\tau(G) : v \in C\}$. Now, consider the following *Reduced Master Problem* (RMP) associated with the linear programming relaxation of CLHF:

Formulation 3. *CLHF - The Reduced Master Problem (RMP)*

$$z_{RMP} =$$

$$\min \sum_{\substack{C \in \mathcal{C}^\tau(G) \\ |C| \geqslant 2}} f(C) y_C + \sum_{v \in V} f(v) x_v \tag{13a}$$

$$\text{s.t.} \sum_{\substack{C \in \mathcal{C}^\tau_v(G) \\ |C| \geqslant 2}} y_C + x_v \geqslant 1 \qquad \forall\, v \in V \tag{13b}$$

$$x_v \geqslant 0 \qquad \forall\, v \in V \tag{13c}$$

$$y_C \geqslant 0 \qquad \forall\, C \in \mathcal{C}^\tau(G), |C| \geqslant 2. \tag{13d}$$

Note that in the RMP constraints $x_v \leqslant 1$, for all $v \in V$, and $y_C \leqslant 1$, for all $C \in \mathcal{C}^\tau(G)$, $|C| \geqslant 2$, have been omitted as they are redundant. The RMP can be solved by column generation techniques. To this end, denote μ_v as the dual variables associated with constraints (13b). The dual of the RMP is characterized by the following dual constraints:

$$\mu_v \leqslant f(v) \qquad \forall\, v \in V \tag{14a}$$

$$\sum_{v \in V : v \in C} \mu_v \leqslant f(C) \qquad \forall C \in \mathcal{C}^\tau(G) : |C| \geqslant 2. \tag{14b}$$

A variable with negative reduced cost to be added to the RMP corresponds to a dual constraint violated by the current optimal solution. Since all of the x_v variables are present in the RMP, constraints (14a) will never be violated. In contrast, constraints (14b) are violated if

$$\exists\, \hat{C} \in \mathcal{C}(G) : |\hat{C}| \geqslant 2 \text{ and } \sum_{v \in V : v \in \hat{C}} \mu_v > f(\hat{C}). \tag{15}$$

Checking whether (15) holds in the current optimal solution to the RMP involves solving a maximum node-weighted clique problem in G with weights $\{\mu\}$. This task can be performed e.g., by means of Algorithm 2 that is described in Section 3.2.2. Algorithm 1 implements the column generation technique by using Algorithm 2 as pricing oracle. Specifically, Algorithm 1 takes G as an input and iterates the following steps: line 2 solves the RMP; line 3 gets the dual values $\{\mu\}$ and

Algorithm 1 | Column Generation

Input: G

Output: Optimal solution to the RMP

1: **repeat**
2: Solve the RMP
3: Get the dual vector μ associated to (13b)
4: Set weight μ_v on each vertex $v \in V$
5: $\hat{C}$ =PricingOracle(G,μ)
6: **if** $\hat{C} \neq \emptyset$ **then**
7: Add $y_{\hat{C}}$ and its column to the RMP
8: **until** $\hat{C} = \emptyset$
9: return z^*_{RMP}

associates a weight μ_v to each vertex $v \in V$; line 5 calls the oracle and searches in $\mathcal{C}(G)$ for a maximum node-weighted clique $\hat{C}$ satisfying (15); if such a clique exists, line 7 adds variable $y_{\hat{C}}$ (and its corresponding column) to the RMP and iterates lines 1-8; if, there are NO violated constraints (13b), the solution to the RMP is provably optimal and Algorithm 1 returns the optimal value to the RMP. Preliminary experiments showed that to decrease the running time of Algorithm 1 it is advisable to add not just the most violated node-weighted clique at each iteration of the pricing oracle, but possibly many cliques C having an overall weight larger than $f(C)$. Section 3.3 explores, via computational experiments, the problem of deciding how many cliques should be added in a generic instance of the PLOHP.

3.2.2 *Finding a maximum node-weighted clique in a MPTG*

A critical step in Algorithm 1 consists of finding a maximum node-weighted clique in G. Although this problem is *NP*-hard for general graphs, it can be solved in polynomial time when the graph is MPTG and the objective function is size-defined submodular. In fact, it is

Algorithm 2 | Pricing Oracle

Input: G, μ

Output: $\mathcal{C}^\star$: a maximum node-weighted clique in G

```
1:  set C = ∅, max = 0, get 𝓘 from G
2:  for all I_i, I_j ∈ 𝓘 : i < j  and  adjacent(I_i, I_j) do
3:      set cost = 0
4:      for all I_k ∈ 𝓘 do
5:          if (p_i ⩽ p_k ⩽ p_j) ∧ (l_k < p_i) ∧ (r_k > p_j) then
6:              C = C ∪ {p_k}
7:      for all v ∈ V : v ∈ C do
8:          cost = (cost + μ_v)
9:      if cost > max then
10:         max = cost
11:         𝒞* = C
12: return 𝒞*
```

worth noting that, as G is a MPTG, any subset of vertices that forms a maximal clique C in G is characterized by having two vertices whose corresponding points, say p_l and p_r, are the furthest to the left and to the right, respectively, in the set of point-intervals induced by C. Each vertex v in the clique is connected to these two vertices and its corresponding point-interval is such that $p_v \in [p_l, p_r] \subseteq [l_v, r_v]$. Hence, a clique can be defined by the leftmost and rightmost vertices, respectively. Then, a possible approach to enumerate the polynomial bounded number of cliques in G consists of picking any pair of adjacent vertices in G (i.e., distinct point-intervals in $\mathcal{I}$, say (I_u, p_u) and (I_v, p_v), such that $I_u \cap I_v = (l_u, r_u) \cap (l_v, r_v) \supseteq \{p_u, p_v\}$) and searching for the largest set of points that fall inside $[p_i = \min\{p_u, p_v\}, p_j = \max\{p_u, p_v\}]$, i.e., the largest set of point-intervals I_k that satisfy condition $(p_i \leqslant p_k \leqslant p_j) \wedge (l_k < p_i) \wedge (r_k > p_j)$. Algorithm 2 outlines the pseudo code necessary to perform the overall task. We note that the complexity of Algorithm 2 is $O(|\mathcal{I}|^3)$.

3.2.3 *Initializing the columns of the RMP*

In order to initialize the RMP it is necessary to find a subset $\mathcal{C}^\tau(G) \subset \mathcal{C}(G)$ covering all of the vertices in G and such that any clique in $\mathcal{C}^\tau(G)$ has cardinality greater than or equal to 2. This is due to the presence of all the x_v variable in the RMP. A possible approach to carry out this task is outlined in Algorithm 3. In particular, the algorithm takes it as input the set $\mathcal{I}$ in non-decreasing order. Subsequently iterates the following procedure: for each point-interval $I_k \in \mathcal{I}$ (i) find a maximal clique C that cover (the vertex induced by) I_k and (ii) remove from $\mathcal{I}$ the point-intervals (whose induced vertices are) in C. The procedure stops when $\mathcal{I} = \emptyset$. The complexity of Algorithm 3 is $O(|\mathcal{I}|^3)$.

Algorithm 3 | $\mathcal{C}^\tau(G)$-Generator

Input: Set of intervals $\mathcal{I}$

Output: $\mathcal{C}^\tau(G)$

1: $\mathcal{C} = \emptyset$, $Q = \mathcal{I}$
2: **for all** $I_i, I_j \in \mathcal{I} : i < j$ **and** adjacent(I_i, I_j) **and** $Q \neq \emptyset$ **do**
3: $\quad C = \emptyset$
4: $\quad$ **for all** $I_k \in \mathcal{I}$ **do**
5: $\quad\quad$ **if** COMPATIBLE(I_i, I_j, I_k) **then**
6: $\quad\quad\quad Q = Q \setminus I_k$
7: $\quad\quad\quad C = C \cup I_k$
8: $\quad \mathcal{C} = \mathcal{C} \cup C$
9: **return** $\mathcal{C}$

10: **function** COMPATIBLE(I_u, I_v, I_z)
11: $\quad$ **return** $(p_u \leqslant p_z \leqslant p_v) \wedge (l_z < p_u) \wedge (r_z > p_v)$

3.2.4 *Branching rules*

The optimal solution to the RMP can be fractional. To ensure its integrality we embody the previously described column generation approach in a B&B algorithm, in which the branching is performed on both x and y variables. Specifically, concerning x variables, the algorithm branches according to the *most fractional variable* rule (see Nemhauser et al. [1989]). The same rule also applies to y variables with the exception that a y variable that has been set to 0 in a previous node of the search tree cannot be set to 1 in one of its descendants. To ensure the application of this rule we perform the following steps: (1) we store all the maximal clique generated in each node; (2) during the column generation process, we filter, from among all of the maximal cliques, the ones associated to the y variables that have been set to 0 in the parent node; (3) we propagate the filtered set to all the descendant nodes.

3.2.5 *Preprocessing*

In this section we develop a number of preprocessing techniques both to reduce the size of an instance of the PLOHP and to decompose an instance into a family of independent subproblems having a smaller size than the original one. We discuss the computational impact of these techniques in Section 3.3.

3.2.5.1 *Removing Vertices*

A basic approach to reduce the size of an instance of the PLOHP consists of removing variables from CLHF whose value can be known a-priori and consequently remove the constraints associated with those variables. This is possible e.g., when dealing with isolated vertices in the graph. In fact, any feasible solution to CLHF is characterized by

having $x_v = 1$ for any isolated vertex $v \in V$. Isolated vertices can be efficiently detected by inspecting the set $\mathcal{I}$ for intervals having no-compatibility between each other. For example, the point-interval $\mathcal{I}_v$ induces an isolated vertex in V if there does not exists in $\mathcal{I}$ a distinct point-interval $\mathcal{I}_u$ such that $(l_u \leqslant p_v \leqslant r_u) \wedge (l_v \leqslant p_u \leqslant r_v)$. It is easy to see that the removal of isolated vertices from an instance of the PLOHP can be performed in $O(|\mathcal{I}|(|\mathcal{I}|-1)/2)$.

Interestingly, isolated vertices are not the only vertices that can be removed from an instance of the PLOHP. For example, given a point-interval $I_v = [(l_v, r_v), p_v] \in \mathcal{I}$, consider the situation in which there exists another interval $I_u = [(l_u, r_u), p_u] \in \mathcal{I}$ such that $l_u = l_v,\ p_u = p_v,\ r_u = r_v$. In such a case, we say that I_u is a *cloned interval* of I_v. Let Γ_{I_v} be the set of cloned intervals of I_v. Then, it is easy to see that we can remove from V the set of vertices induced by Γ_{I_v} and keep just I_v as representative of the induced equivalence class Γ_{I_v}. The identification of the cloned intervals can be performed in time $O(|\mathcal{I}|(|\mathcal{I}|-1)/2)$ and can be carried out together with the removal of isolated vertices.

3.2.5.2 *Decomposing a MPTG into connected components*

The particular nature of point-intervals may give rise to MPTG with several connected components. For example, the point-intervals shown in Figure 8 leads to to a MPTG with by two connected components (see Figure 9). The potential existence of connected components in a MPTG can be exploited both to reduce the time necessary to identify a maximal clique in G and to decompose the problem into independent subproblems of smaller size. A possible way to quickly identify connected components in a MPTG consists of ordering the point-intervals according to a specific precedence rule. In particular, given two distinct point-intervals $I_u, I_v \in \mathcal{I}$, we say that I_v precedes I_u, in symbols $I_v \prec I_u$, if one of the following cases is verified:

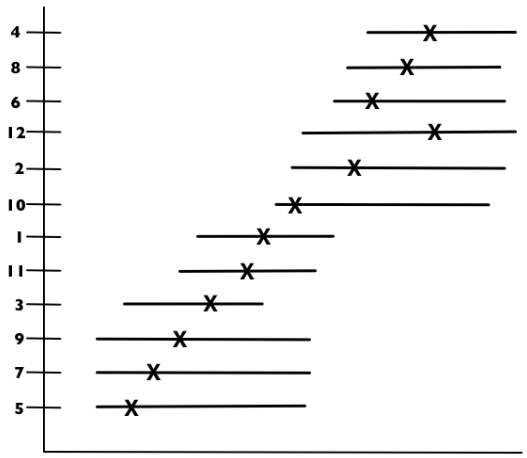


Figure 10: An example of block decomposition of the point-intervals shown in Figure 8.

1. $l_v < l_u$;
2. if $l_v = l_u$ then $p_v < p_u$;
3. if $l_v = l_u$ and $p_v = p_u$ then $r_v \leqslant r_u$.

For example, Figure 10 shows the result of the application of the precedence rule on the point-intervals shown in Figure 8. It is easy to realize that, if the point-intervals have been sorted according to the above precedence rule, a connected component arises in G whenever there exist two $I_u = \{(l_u, r_u), p_u\}$ and $I_v = \{(l_v, r_v), p_v\}$ such that $p_v \leqslant l_u$.

Algorithm 4 provides a heuristic to identify connected components in a MPTG. In fact condition $p_i < l_{i+1}$ is sufficient to identify a set of nodes not connected with a given node, but not necessary. We opted for a heuristic approach rather then an exact one because in preliminary experiments we observed that the exact approach was slower than the heuristic one and did not prove particularly beneficial in terms of identification of connected components. The algorithm takes as input the set $\mathcal{I}$ sorted according to the above precedence rule. Subsequently, the algorithm visits each point-interval in $\mathcal{I}$ and constructs a connected component B of G by collecting all of the point-intervals

I_i such that I_i does not satisfy the condition $p_i \leqslant l_{i+1}$. It is easy to see that the computational complexity of Algorithm 4 is $O(|\mathcal{I}|)$. Moreover, it is worth noting that the decomposition of G into connected components enables the parallelization of the RMP, an approach that may vastly shorten the solution time of an instance of the problem.

3.3 EXPERIMENTAL RESULTS

In this section we analyze the performance of the B&P algorithm to solve instances of the parsimonious loss of heterozigosity problem. Our experiments were motivated by a number of goals, namely: to compare the performance of the B&P algorithm with the ones obtained by CLHF [Catanzaro et al. 2013], which currently is the best exact algorithm for the PLOHP; to evaluate the benefits obtained by using the presolving and block decomposition strategies previously described; and finally, to allow the analysis of larger data sets with respect to the ones currently analyzed. As in Catanzaro et al. [2013] and Halldórsson et al. [2011b], we emphasize that our experiments simply aim to evaluate the computational performance of the exact algorithms. We neither attempt to study the efficiency of the B&P al-

Algorithm 4 | Connected Components Finder

Input: sorted $\mathcal{I}$

Output: $\mathcal{K}$: set of connected components of G

1: $i = 0$, $K = \emptyset$, $\mathcal{K} = \emptyset$
2: **while** $i < |\mathcal{I}|$ **do**
3: **if** $p_i > l_{i+1}$ **then**
4: $K = K \cup \{I_i\}$
5: **else**
6: $\mathcal{K} = \mathcal{K} \cup \{K\}$, $K = \emptyset$
7: **return** $\mathcal{K}$

gorithm to predict LOH events across the genomes of a population of individuals nor to compare its accuracy versus LOH predictors that do not use the parsimony criterion. This analysis has been performed by Halldórsson et al. [2011b], Conrad et al. [2006], and Speed et al. [2007] and McCarroll et al. [2008] and we refer the interested reader to these articles for further information.

3.3.1 *Implementation*

We implemented CLHF in ANSI C++ by using FICO Xpress 7.6, Optimizer libraries v26.01.04 (64-bit Hyper capacity). We implemented the B&P algorithm in ANSI C++ by using SCIP Optimization Suite 3.1.0 [Achterberg 2009] to handle the column generation and the B&P routines. Moreover, we used FICO Xpress Optimizer as linear programming solver. The experiments have been performed on an Intel Core i7-4930K CPU, 3.40GHz, equipped with 64 GByte RAM and operating system Ubuntu release 12.10 (kernel linux 3.5.0-41-generic). During the runtime of CLHF we activated Xpress automatic cuts, Xpress presolving strategy, and Xpress primal heuristic to generate the first upper bound for the PLOHP. Finally, similar to Halldórsson et al. [2011b] and Catanzaro et al. [2013], we set the maximum runtime for the Formulations to 5 hours. The executable codes used in the experiments can be downloaded together with the instances at url: `http://homepages.ulb.ac.be/~lporrett/PLOHP/Code.tar.bz2`.

3.3.2 *Calibrating the pricing oracle*

Determining the number of variables with negative reduced cost to be added in the RMP is crucial for the performances of the proposed B&P. To perform this task, we generated a learning set of instances of the PLOHP constituted by 100 random instances. We generated

	Nodes	Edges	Cliques
Maximun	66 670	2 060 044 865	52 522
Minimum	35 725	742 125 353	16 915
Average	57 581.35	1 335 575 876.85	36 824.55
Standard Deviation	8 056.94	313 44 967.49	8 011.84

Table 6: Summary of number of nodes, edges and cliques contained in the learning set of instances of the PLOHP used to calibrate the pricing oracle.

such a set by using the same approach and parameters discussed in Halldórsson et al. [2011b]. Each random instance has between 35725 and 66670 point-intervals and between 16915 and 52522 cliques on the real segment $[1, 1000]$. Table 6 reports on the main characteristics of this set. We introduced a parametrized number of violated cliques to be added by the pricing oracle at runtime. Such a number is a fraction of the overall number of maximal cliques contained in a random instance and we set it to be equal to one of the following seven different percentages: 1%, 3%, 5%, 7%, 10%, 15% and 20%. Figure 11 shows the box-and-whisker plot histogram of the solution time taken by the B&P algorithm to exactly solve the learning set of instances of the PLOHP when considering different percentages of violated cliques. A box shows the range between the 25% and the 75% quantile of the data. The median of the data is indicate by a bar. The whiskers extend to the most extreme data point which is no more than 1.5 times the interquantile range from the box. The Figure shows that the median of the solution time taken by the B&P algorithm to exactly solve the learning set of instances of the PLOHP is minimum when adding in the RMP no more than 1% of violated cliques. To exclude the presence of statistical equivalence between the solution times in correspondence to different percentages of violated cliques, we run a Wilcoxon signed-

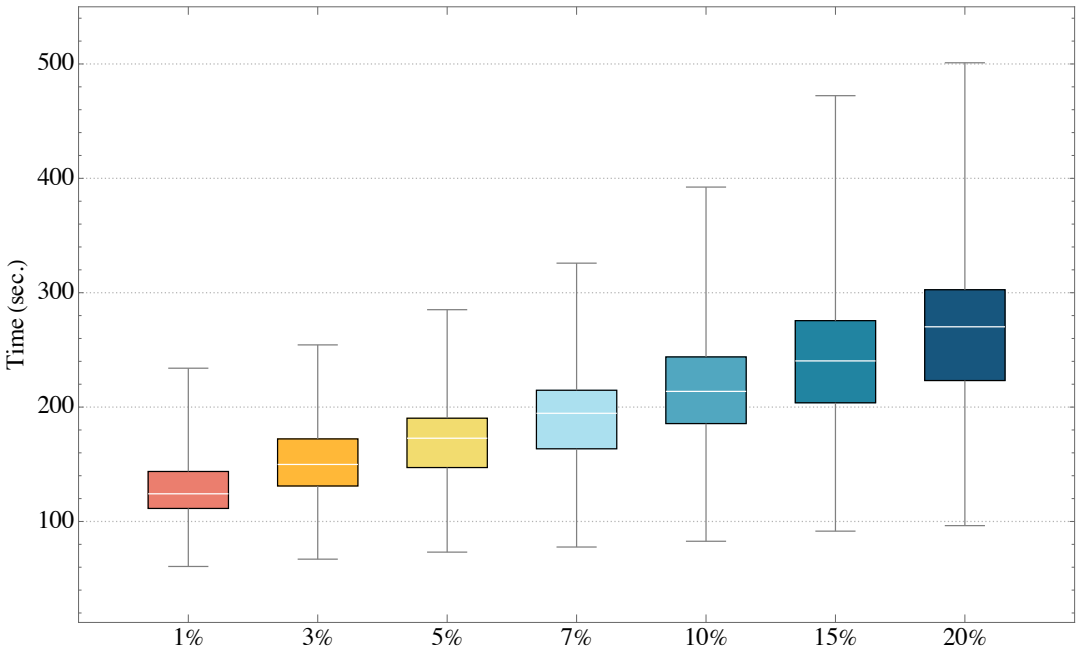


Figure 11: BoxPlot of the solution time taken by the B&P algorithm to exactly solve the learning set of instances of the PLOHP when considering different percentages of violated cliques.

rank test with Holmes correction for multiple tests [Fay et al. 2010]. Table 7 reports on the p-values so obtained. Specifically, the presence of a value smaller then 0.05 in a cell means that the sample in the row is statistically significant smaller than the sample in the column. In contrast, the presence of a value bigger than 0.05 in a cell means that the sample in the row is statistically significant bigger than the sample in the column. In particular, Table 7 shows that adding the 1% of the violated cliques always provide a smaller solution time compared to the others approaches. For this reasons, we decided to use the setup in applying the B&P algorithm to the remaining experiments. For the sake of completeness, we should say that we also performed tests using others criteria for selecting the variables to add at the RMP. In particular, we tried to add at each iteration only the most violated clique but the performance results to be really poor. We also tried to randomly select one violated clique at each iteration avoiding the sort-

ing of the variables by their negative reduced cost, but unfortunately also these test performed poorly.

3.3.3 *Benchmark instances*

In order to compare the performance of the B&P algorithm with the ones obtained by CLHF [Catanzaro et al. 2013], we considered the instances of the PLOHP described in Catanzaro et al. [2013] and Halldórsson et al. [2011b]. These instances, hereafter denoted as "gens-1", "gens-2", "gens-3", "gens-4" and "gens-5", are characterized by having 645572, 645552, 645510, 645543 and 659885 point-intervals on the real segment $[1, 3575]$, respectively, each having minimum length 1 and maximum length 9. Catanzaro et al. observed that the ILP formulation presented in Catanzaro et al. [2013] proved unable to solve any of these instances within 1 hour of computing time. Hence, the authors also considered smaller instances of the PLOHP obtained from the given ones by extracting from each "gens-x", $x \in \{1, 5\}$, the point-intervals contained in the real segments $\alpha = [1, 1500]$, $\beta = [1, 2000]$ and $\gamma = [1, 3575]$, respectively. The authors obtained this way 15 instances of the PLOHP, that are hereafter grouped into three datasets denoted as "gens-x_α", "gens-x_β" and "gens-x_γ", $x \in \{1, 5\}$, respec-

	1%	3%	5%	7%	10%	15%	20%
1%	—	1.97784×10^{-18}	1.97784×10^{-18}	1.97784×10^{-18}	1.97796×10^{-18}	1.97796×10^{-18}	1.97796×10^{-18}
3%	1	—	2.95093×10^{-17}	1.97796×10^{-18}	1.97796×10^{-18}	1.97784×10^{-18}	1.97796×10^{-18}
5%	1	1	—	1.79421×10^{-17}	2.03865×10^{-18}	1.97796×10^{-17}	1.97784×10^{-18}
7%	1	1	1	—	6.37311×10^{-18}	1.97784×10^{-18}	1.97796×10^{-18}
10%	1	1	1	1	—	4.83732×10^{-17}	2.16549×10^{-18}
15%	1	1	1	1	1	—	2.4767×10^{-18}
20%	1	1	1	1	1	1	—

Table 7: Wilcoxon signed-rank test on solution time for each pair of percentages

	Reference	Overall point Intervals	Edges	Connected Components	Cloned point Intervals	Singletons
gens-1_α	[1,1 500]	84 435	49 690 310	29	59 666	3
gens-2_α	[1,1 500]	84 436	49 681 938	29	59 659	0
gens-3_α	[1,1 500]	84 427	49 669 238	30	59 669	0
gens-4_α	[1,1 500]	84 440	49 660 650	29	59 656	5
gens-5_α	[1,1 500]	86 467	50 023 009	37	58 052	5
gens-1_β	[1,2 000]	245 205	289 645 236	37	191 817	4
gens-2_β	[1,2 000]	245 198	289 599 050	39	191 832	4
gens-3_β	[1,2 000]	245 177	289 589 450	40	191 907	3
gens-4_β	[1,2 000]	245 197	289 527 061	38	191 904	7
gens-5_β	[1,2 000]	250 539	291 137 895	33	187 199	4
gens-1_γ	[1,3 575]	645 572	1 137 465 967	61	526 689	13
gens-2_γ	[1,3 575]	645 552	1 137 474 258	65	526 688	9
gens-3_γ	[1,3 575]	645 510	1 137 451 358	62	526 657	11
gens-4_γ	[1,3 575]	645 543	1 137 258 659	62	526 703	13
gens-5_γ	[1,3 575]	659 885	1 143 798 666	40	514 953	2

Table 8: Statistics of gens-x_α , gens-x_β , gens-x_γ

tively.

Table 8 shows the main characteristics of the 15 instances. In particular, the table is composed by three main sections, each one corresponding to the datasets "gens-x_α", "gens-x_β" and "gens-x_γ", $x \in \{1,5\}$, previously described. The first three columns of Table 8 report the size of the reference segment, the overall number of point-intervals and edges contained in a given instance. The last three columns report the number of connected components, cloned intervals and singletons contained in a given instance. Table 8 shows that the instances "gens-x_α", $x \in \{1,5\}$ are characterized by a number of connected components ranging from 29 to 37, a number of single-

Set of Instances	Point-interval Length	Max. number of Point-intervals	Min. number of Point-intervals	Average number of Point-intervals
Sim-1	2	23963	23777	23851 ±52.79
Sim-2	5	24012	23823	23950.5±58.03
Sim-4	7	23615	23454	23553.5±50.67
Sim-3	9	23777	23803	23828 ±42.55
Sim-5	random in [1,9]	29020	28699	28904 ±94.11

Table 9: Parameter sets used to generate random instances of the PLOHP.

tons ranging from 0 to 5 and over 70% of cloned point-intervals. The instances "gens-x_β", $x \in \{1,5\}$ are characterized by a number of connected components ranging from 33 to 40, a number of singletons ranging from 3 to 7 and over 70% of cloned point-intervals. Finally, the instances "gens-x_γ", $x \in \{1,5\}$ are characterized by a number of connected components ranging from 40 to 61, a number of singletons ranging from 2 to 13 and over 80% of cloned point-intervals.

We also considered a set of 5 artificial datasets of the PLOHP, hereafter denoted as "Sim-1", "Sim-2", "Sim-3", "Sim-4" and "Sim-5", each containing 10 random instances of the problem and mainly differing from one another both by the length and the number of the point-intervals contained in each of the corresponding instances. We generated the instances in each dataset by first fixing the real segment $[1, 6000]$ as a reference and by generating, by means of the Mersenne Twister library [Matsumoto et al. 1998], a random number of triplets (l_v, p_v, r_v) such that $1 \leqslant l_v \leqslant p_v \leqslant r_v \leqslant 6000$. Table 9 summarizes the characteristics of the considered datasets. Finally, in order to evaluate the performance of the B&P algorithm also on biological instances of the PLOHP, we considered a biological instance provided by deCode Genetics and hereafter called "Bio22". This in-

stance is constituted by 20022 point-intervals on the real segment $[1, 6000]$, each having minimum length 2 and maximum length 6000. The point-intervals in Bio22 refers to genetic fragments from chromosome 22 extracted from a population of 18360 individuals and constists of 6000 single nucleotide polymorphisms. All the instances used in our experiments can be downloaded at `http://homepages.ulb.ac.be/~lporrett/PLOHP/Datasets.tar.bz2`.

3.3.4 *Computational performances*

Table 10 shows the performances of CLHF and the B&P in solving the 5 real instances described in Halldórsson et al. [2011b] with and without the presolving strategies. In particular, the table is constituted by three main sections, each one corresponding to the datasets "gens-x_α", "gens-x_β" and "gens-x_γ", $x \in \{1, 5\}$, previously described. Moreover, the columns are divided in four sections, each of them showing the solution time (expressed in seconds) and the number of cliques needed to optimally solve a given instance of the PLOHP. Specifically, the first pair of columns is related to the execution of CLHF. The second pair of columns is related to the execution of the B&P. The third pair of columns is related to the execution of CLHF using the presolving strategies. Finally, the fourth pair of columns are related to the execution of the B&P using the presolving strategies. As a general trend, Table 10 shows that, independently of the presence (or absence) of the presolving strategies, the B&P is always faster than CLHF. This fact alone justifies and shows the importance of using inverse projection when dealing with very large instances of the PLOHP. Moreover, the table also shows the benefits of using the presolving strategies: the removal of over 70% of cloned intervals allows the removal of over 70% of x_ν variables; the search for the connected components enables the decomposition of the corresponding induced MPTG into at least

	CLHF		B&P		CLHF + Preprocessing		B&P + Preprocessing	
	Time (sec.)	Cliques	Time (sec.)	Cliques	Time (sec.)	Cliques (Avg.)	Time (sec.)	Cliques (Avg.)
gens-1_α	1 062.46	63660	316.99	1 845	70.83	2 490.00± 3 684.44	63.06	587.59 ±1 410.13
gens-2_α	1 145.84	63784	302.88	1 859	68.14	2 497.38± 3 686.79	58.91	256.62 ± 563.91
gens-3_α	1 090.85	63303	312.52	1 845	63.42	2 288.30± 2 491.55	56.54	166.97 ± 166.97
gens-4_α	1 035.16	63507	301.02	1 871	67.67	2 421.41± 2 719.01	62.13	551.07 ± 877.04
gens-5_α	1 198.22	69266	378.390	2 852	74.19	2 055.49± 2 118.55	68.93	551.07 ± 193.30
gens-1_β	9 039.54	148 862	2 455.14	3 532	350.39	5 095.03± 9 235.71	314.69	1 267.97±4 578.73
gens-2_β	8 965.42	148 505	2 743.00	3 521	313.93	4 843.21± 7 851.87	267.17	534.77 ±1 338.22
gens-3_β	8 901.90	148 042	2 562.65	3 510	292.18	4 691.43± 7 651.59	258.19	305.28 ± 756.50
gens-4_β	8 925.68	147 249	2 630.90	3 508	328.28	4 909.92± 7 926.41	279.21	797.00 ±2 868.75
gens-5_β	9 488.87	138 627	3 805.31	5 739	500.18	5 033.58± 7 523.26	396.62	906.76 ±2 107.23
gens-1_γ	>5h	—	>5h	—	2 005.03	7 219.23±15 025.36	1 800.91	752.97 ±2 334.55
gens-2_γ	>5h	—	>5h	—	1 901.62	6 939.11±14 928.34	1 708.04	525.54 ±1 390.90
gens-3_γ	>5h	—	>5h	—	1 973.48	7 227.92±15 273.65	1 757.70	604.76 ±1 881.98
gens-4_γ	>5h	—	>5h	—	1 978.90	7 065.68±15 232.85	1 754.85	252.03 ± 579.87
gens-5_γ	>5h	—	>5h	—	3 292.84	9 311.35±11 355.82	2 859.05	437.95 ± 635.70

Table 10: Computational performances comparison of CLHF and the B&P algorithm with and without presolving strategies

29 connected components (independent subproblems), which leads to a vast speed-up of the solution process. Finally, the table shows that, although the B&P is faster than CLHF, it proves unable to solve the instances "gens-x_γ", $x \in \{1, 5\}$, within the considered time limit. This fact is mainly due to the pricing oracle, which proves particularly time-consuming in gens-x_γ.

3.3.5 *Computational performances on larger datasets of the PLOHP*

Table 11 shows the performance of CLHF and the B&P in solving the "Sim" instances when using the presolving strategies. Both CLHF and the B&P algorithm without presolving strategies proved unable to solve any instance in this set. The table reports on the mean and standard deviation of: the solution time (expressed in seconds) necessary to solve a random instance of the PLOHP; the gap (expressed in

	CLHF			B&P		
Instance	Time (sec.)	Gap (%)	Nodes	Time (sec.)	Gap (%)	Nodes
Sim-1	1 217.965±41.14918	0.0107±0.1838	1.0127±0.2001	577.7767±18.24831	0.0039±0.1350	1.0033±0.1213
Sim-2	1 200.470±39.02218	0.0096±0.1770	1.0108±0.1871	552.4411±40.46290	0.0040±0.1209	1.0042±0.1160
Sim-3	1 200.242±48.61129	0.0072±0.1318	1.0107±0.1939	575.2736±32.15120	0.0010±0.0582	1.0012±0.0699
Sim-4	1 162.675±94.31992	0.0078±0.1453	1.0129±0.2697	509.0114±84.75060	0.0033±0.0941	1.0037±0.08602
Sim-5	**OOM±OOM**	**OOM±OOM**	**OOM±OOM**	969.5657±424.2924	0.0026±0.0951	1.0019±0.06258

Table 11: Computational performances comparison of CLHF and the B&P algorithm on artificial instances of the PLOHP.

percentage), i.e., the difference between the optimal value to a given instance of the PLOHP in a specific dataset and the objective function value of the linear programming relaxation at the root node of the respective search tree, divided by the optimal value; the nodes explored in the search tree respectively.

In general, Table 11 shows that CLHF and the B&P, combined with presolving strategies, is able to solve each simulated instance in about half an hour, except for the parameter set "Sim-5" that required on average 969.5657 seconds for the B&P while CLHF exceeds the 64GB of memory available for the experiments.

Tables 12 report on the minimum, maximum and average number of connected components,intervals, singletons, edges and Cloned nodes, while Table 13 report on the minimum, maximum and average time (expressed in seconds) to filter the singletons and the cloned intervals, and to identify the connected components.

Table 14 shows the performance of the B&P in solving "Bio22" instance using the presolving strategies. The table reports the solution time (expressed in seconds) necessary to solve the biological instance of the PLOHP; the gap (expressed in percentage), i.e., the difference between the optimal value and the objective function value of the linear

	Sim1			**Sim2**		
	Min	*Max*	*Average*	*Min*	*Max*	*Average*
# Conn. Comp.	293	311	298	291	307	295
# Intervals	20 562	20 763	20 689	205 270	20 693	20 587
# Singletons	18	32	24	16	30	26
# Edges	2 824 349	2 922 094	2 881 541	281 133	2 889 021	2 854 612
# Cloned Nodes	3 218	3 243	3 237	3 223	3 255	3 241
	Sim3			**Sim4**		
	Min	*Max*	*Average*	*Min*	*Max*	*Average*
# Conn. Comp.	288	303	298	279	305	293
# Intervals	20 539	20 670	20 567	20 187	20 359	20 312
# Singletons	16	30	23	18	31	26
# Edges	2 810 257	2 862 659	2 833 031	2 728 854	2 767 026	2 744 176
# Cloned Nodes	3 222	3 254	3 242	3 221	3 241	3 233
	Sim5					
	Min	*Max*	*Average*			
# Conn. Comp.	259	291	28			
# Intervals	25 328	25 649	25 525			
# Singletons	5	16	11			
# Edges	4 688 361	4 810 792	4 760 66			
# Cloned Nodes	3 358	3 391	3 371			

Table 12: Statistics of simulated instances Sim1, Sim2, Sim3, Sim4, Sim5.

programming relaxation at the root node of the search tree, divided by the optimal value; the nodes explored in the search tree; and the overall number of connected components, intervals, cloned intervals and singletons, respectively.

Table 11 also shows that the solution time necessary to solve the artificial instances of the PLOHP is longer than the corresponding one necessary to solve any other instances. This fact is possibly due to

	Sim1			**Sim2**		
	Min	*Max*	*Average*	*Min*	*Max*	*Average*
Singleton	0.43	0.52	0.44	0.43	0.52	0.43
Clone	0.42	0.43	0.42	0.42	0.50	0.42
Conn. Comp.	$6.2{\cdot}10^{-5}$	$1.3{\cdot}10^{-5}$	$6.2{\cdot}10^{-5}$	$6.1{\cdot}10^{-5}$	$1.7{\cdot}10^{-5}$	$6.2{\cdot}10^{-5}$
	Sim3			**Sim4**		
	Min	*Max*	*Average*	*Min*	*Max*	*Average*
Singleton	0.43	0.52	0.43	0.41	0.42	0.41
Clone	0.42	0.50	0.42	0.40	0.41	0.42
Conn. Comp.	$6.1{\cdot}10^{-5}$	$6.2{\cdot}10^{-5}$	$6.2{\cdot}10^{-5}$	$6.0{\cdot}10^{-5}$	$1.3{\cdot}10^{-5}$	$6.1{\cdot}10^{-5}$
	Sim5					
	Min	*Max*	*Average*			
Singleton	0.73	0.78	0.74			
Clone	0.65	0.66	0.65			
Conn. Comp.	$7.7{\cdot}10^{-5}$	$7.9{\cdot}10^{-5}$	$7.8{\cdot}10^{-5}$			

Table 13: The minimum, the maximum and the average time (expressed in seconds) both to filter the singletons and the cloned intervals, and to identify the connected components in Sim1, Sim2, Sim3, Sim4, Sim5.

Instance	Time (sec.)	Gap (%)	Nodes	Connected Components	Point Intervals	Cloned point Intervals	Singletons
Bio22	661.40	0	1	321	19968	3151	54

Table 14: Computational performances of the B&P on biological instance of the PLOHP.

the larger number of point-intervals present in the artificial instances which in turn increases the number of cliques in the corresponding MPTG.

In general, Table 14 shows that the B&P, combined with presolving strategies, is able to solve the biological instance in 664.40 seconds. Interestingly, Table 11 shows that the biological instance differs from the instances in Table 10 by having a fewer number of point-intervals and a relatively smaller number of cloned ones. In this case, filtering the cloned point-intervals reduces the number of x_v variables only by the 20%, instead of the reduction of variables x_v by 70% shown in Table 10. However, the relatively higher number of connected components in Bio 22, allows to decompose the problem into a bigger family of small independent subproblems, providing a remarkable speed up in the solution time.

In the next chapter we study a problem arising while analysing the whole genome re-sequencing. Instead of having only the information about the SNPs genotype, the whole genome data allows the detection of all kinds of genomic variation and not only the ones highlighted by the SNPs. In particular, we examine the polymorphism present in the human genome that is caused by the insertion/deletion of a specific family of copy-number variations.

4

AN INTEGER PROGRAMMING FORMULATION OF THE MULTIPLE INDIVIDUAL POLYMORPHIC ALU INSERTION RECOGNITION PROBLEM

Whole genome re-sequencing using next generation sequencers rapidly become the sledgehammer of genome wide association studies and started to be preferable over SNP genotyping for association studies as it allows for the detection of all genomic variation and not only SNP variation. Even if SNPs remains the most abundant form of variation between two individuals, other forms of variation exist, such as inversions, copy-number variations, *Long INterspersed Elements* (LINE) and *Short INterspersed Elements* (SINE) elements.

In this chapter we analyze a particular form of SINE variation called *Arthrobacter Luteus* (ALU) insertions.

An ALU sequence is an approximately 300 basepair long sequence derived from 7SL RNA gene [Ullu et al. 1984]. Although ALU elements do not encode genes, many studies suggest their functional importance. ALU elements are recognized to affect protein synthesis at the transcriptional and post-transcriptional level [Sorek et al. 2002; Kelley et al. 2014] as well as DNA methylation [Andrade et al. 2011] and other cellular processes [Prescott et al. 1999]. Furthermore, they are thought to be major drivers of genome evolution [Hormozdiari et al. 2013; Salem et al. 2003] and assist in the creation of structural variation [Wang et al. 2006]. The importance of ALU elements is further highlighted by the potential association with genetic instability, one of the principal cause factors in many disorders including cancer

[Prescott et al. 1999; Zhang et al. 2011; Helman et al. 2014].

ALU repeats are SINE that occur frequently in the human genome, as well as in other genomes. The ALU sequence family has been propagated to more than one million copies in primate genomes over the last 65 million years. ALU repeats are the largest family of mobile elements in the human genome and the ALU family comprises more than 10% of the human genome [Cordaux et al. 2009]. Most ALU repeats were inserted early in primate evolution, where it is estimated that there was approximately one new ALU insertion in every primate birth [Mark et al. 2002].

The current rate of ALU insertion is estimated to be of the order of one ALU insertion in every 200 births [Prescott et al. 1999]. Some members of these young ALU subfamilies have been inserted into the human genome so recently that they are polymorphic with respect to the presence or absence of insertion in different human genomes. Those relatively few elements that are present in the genomes of some individuals and absent from others are referred to as ALU-insertion polymorphisms.

In this chapter we consider the problem of detecting polymorphic ALU insertions from DNA sequence reads using high throughput paired-end sequencing data [Fullwood et al. 2009].

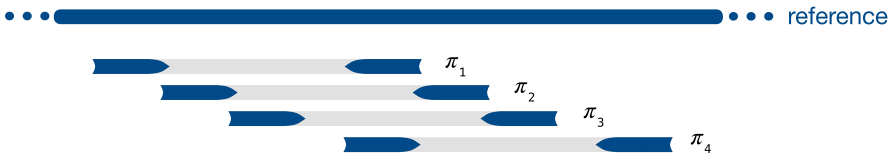
The problem was presented for the first time in Sveinbjörnsson et al. [2012], where the authors provided a polynomial algorithm able to detect ALU insertions coming from a single individual. Simultaneously, they introduce the problem for detecting inserted ALUs over multiple individuals providing an heuristic approach but not an efficient and exact solution method. Specifically, we investigate in detail the problem arising in the multiple individuals scenario by analyzing the combinatorial structure of the problem and presenting a more

general predictive model able to solve it efficiently. In particular we show that the problem of detecting ALU insertion over multiple individuals can be formulated as a specific version of the clique partition problem in a particular class of graphs called *interval graphs* with particular properties. We prove that this problem is *NP*-hard in general and we provide a methodology to solve it based on *Integer Programming* (IP). Specifically, we present an IP formulation and strengthening valid inequalities to reduce the solution space. The work thus gives perspective on the mathematics of detecting ALU insertion in genotype data, provides for the first time methodology suitable for provably optimal solution of hard real instances, and suggests new directions on the development of future efficient exact solution approaches.

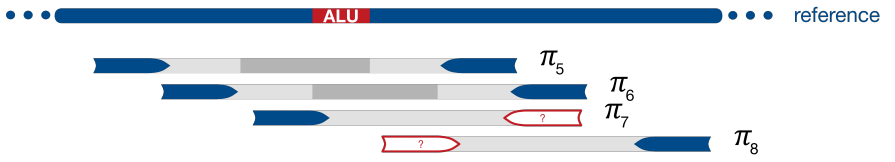
4.1 NOTATION AND PROBLEM STATEMENT

In this section, we introduce notation that will prove useful to state the problem of detecting ALU insertions. In particular, we first introduce a single individual scenario, i.e, insertions detected from DNA sequence reads coming from a single individual. Then, we extend the concepts to the general case in which multiple individuals share the same insertion.

To this purpose, consider an individual as a set $\mathcal{P}$ of read pairs. A read pair consists of a read of a fixed length L, followed by a short spacing Y, followed by another read of the same length L. In particular, the two reads are substrings of DNA sequence, with one read coming from the forward orientation of the strand (from left to right), hereafter called *left read*, and the other coming from the reverse complemented orientation of the strand (from right to left), hereafter called *right read*. The fact that the two reads are read in opposite di-



(a) proper read pairs



(b) improper read pairs

Figure 12: Example of improper read pairs aligned to a reference. Arrows show read directions. The blue part of the reads can be mapped to the reference and the red parts are clipped or mapped somewhere else in the reference. The grey part represents the read pair distance Y and the dark grey part represents the area where the read pair distance Y of the two reads overlaps. The interval of possible ALU locations is highlighted in red over the reference. Note that read pair π_5 and π_6 contribute with two improper reads, while π_7 and π_8 just with one.

rection ensures that: if the location of one of the reads is known then the location of the mate (the other read) is also known, up to Y (see Figure 12a). The spacing between the two reads can be assumed to be known a priori or to be easily estimated from the sequence reads [Korbel et al. 2009]. In particular, let define Y_{max} as the maximal insert length between two reads.

Using the description provided above, we can define a read pair π as a pair of left read π^l and right read π^r, which are mates to each other, denoted as $\texttt{mate}(\pi^l) = \pi^r$ and $\texttt{mate}(\pi^r) = \pi^l$. We use π^n to denote either a left or a right read when the relative position in the pair is not relevant. Since the genome sequence of the individual(s) being sequenced is however not known a priori, but is highly similar to the reference genome, reads are mapped to the reference to infer the genome of the individual. In particular, if π^n is mapped

to a reference genome, we use $\texttt{begin}(\pi^n)$ and $\texttt{end}(\pi^n)$ to represent its start and end position in the mapped reference genome. We define the insert length of a read pair π, measured with respect to the reference genome, as $\Delta(\pi) = \texttt{end}(\pi^r) - \texttt{begin}(\pi^l)$. At some locations in the reference genome the genomes of the reference and the individual(s) being sequenced will diverge. Some of this divergence is due to the insertion of ALU polymorphisms. In particular, whenever $\Delta(\pi) > 2 \times L + Y_{max}$, a read pair is defined as *improper*, i.e., the ends of the two reads map to locations that are inconsistent with the read pair distance Y_{max}. Improper read pair is one of the main signals indicating the presence of an ALU insertion and each left and right read gives partial information about the location of the ALU insertion [Qian et al. 2015].

Specifically, $\delta(\pi^l) = \{p : \texttt{begin}(\pi^l) \leqslant p \leqslant \texttt{end}(\pi^l) + Y\}$ and $\delta(\pi^r) = \{p : \texttt{begin}(\pi^r) - Y \leqslant p \leqslant \texttt{end}(\pi^r)\}$ identify sets of locations in the reference genome candidate to contain an ALU insertion for a left and right read, respectively. By considering two reads π_i^n and π_j^n coming from two distinct improper read pairs i and j, we say that π_i^n and π_j^n are *compatible with an ALU insertion* whenever $\delta(\pi_i^n) \cap \delta(\pi_j^n) \neq \emptyset$ (see Figure 12b). We can now define a *Region Compatible with Insertions* (RCI) as follows:

Definition 1. *Given the set of location of a reference genome $\mathcal{G}$ and $\bar{\mathcal{P}} \subseteq \mathcal{P}$ a set of improper read pairs. Le be $\Pi = \delta(\pi_i^l) \cup \delta(\pi_i^r)$ such that $\pi_i^l, \pi_i^r \in \bar{\mathcal{P}}$, then a Region Compatible with Insertions is any subset $\mathcal{X} \subseteq \mathcal{G}$ such that $\mathcal{X} = \bigcap_{j \in \Pi} \delta(\pi_j^n)$ and $\mathcal{X} \neq \emptyset$.*

Since reads mapping to ALU sequences will almost always have multiple places on the genome that have similar quality mapping, each read either gives evidence of an ALU insertion or to reads improperly mapped or read. In order to detect actual ALU insertions from false positives, denote $\mathcal{R}_i$ and $\mathcal{E}_i$ as the set of RCIs and the set of locations leading to errors induced by individual i respectively, and

$h(\rho)$ and $g(\eta)$ represent the costs of detecting an RCI $\rho \in \mathcal{R}_i$ and a sequencing error $\eta \in \mathcal{E}_i$, respectively. In particular, we can define the following optimization problem:

Problem 3 (*Polymorphic ALU Insertions Recognition Problem* (PAIRP))**.** *Given $\bar{\mathcal{P}}_i$ a set of improper read pairs associated to individual i, minimize the overall cost*

$$\mathcal{Z} = \sum_{\rho \in \mathcal{R}_i} h(\rho) + \sum_{\eta \in \mathcal{E}_i} g(\eta)$$

such that each $\rho \in \bar{\mathcal{P}}$ is either compatible with an RCI or is classified as an error.

A possible extension of the PAIRP to a multiple individuals scenario MIPAIRP, begins by considering that two read pairs $\pi_i \in \mathcal{P}_a$ and $\pi_j \in \mathcal{P}_b$, coming from two different individuals a and b, provide locations compatible with insertion whenever they are mutually compatible. In particular, let denote $\mathcal{R} = \bigcup_{i=1,\dots,n} \mathcal{R}_i$ and $\mathcal{E} = \bigcup_{i=1,\dots,n} \mathcal{E}_i$ as the set of RCIs and the set of locations leading to errors induced by n individuals respectively. We can define the following optimization problem:

Problem 4 (*Multiple Individuals Polymorphic ALU Insertions Recognition Problem* (MIPAIRP))**.** *Given $\bar{\mathcal{P}}$ a set of improper read pairs, minimize the overall cost*

$$\mathcal{Z} = \sum_{\rho \in \mathcal{R}} h(\rho) + \sum_{\eta \in \mathcal{E}} g(\eta)$$

such that each $\rho \in \bar{\mathcal{P}}$ is either compatible with an RCI or is classified as an error.

The PAIRP and MIPAIRP are based on the parsimony principle [Victor 2005]. This fact implies that the optimal solutions to the problem provide estimations of insertion events that, in the worst case, are lower bounds on the overall number of true insertion events occurred in the set of individuals being considered [Catanzaro 2009, 2011]. Sveinbjörnsson et al. [2012] provided a polynomial algorithm for the PAIRP

and proposed a heuristic approach for the MIPAIRP without investigating the complexity of the latter any further. In the next sections we shall address this major issue and provide an algorithm able to exactly solve instances of both general problems.

4.2 THE ALU GRAPH

In this section we define a new class of graphs, which we term *ALU Graph* (ALUG) and will turn out to be useful in transforming the PAIRP, and by extension the MIPAIRP, into a particular version of the *Minimum Clique Partition Problem* (MCPP) [Michael et al. 1979].

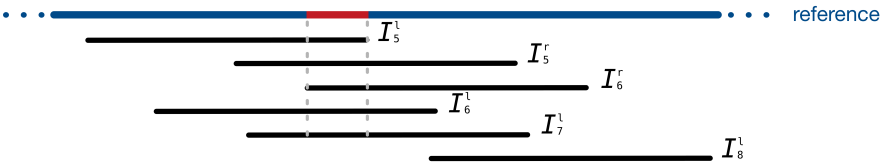
In order to characterize such a class of graphs, consider a set of improper read pairs $\bar{\mathcal{P}}$ and denote $I_k^l = [\texttt{begin}(\pi_k^l), \texttt{end}(\pi_k^l) + Y_{max}]$ and $I_k^r = [\texttt{begin}(\pi_k^r) - Y_{max}, \texttt{end}(\pi_k^r)]$ as the intervals induced by the left and right reads of the k-th read pair, respectively. Moreover, denote $I = I^l \cup I^r$ as the set of intervals induced by all the reads in $\bar{\mathcal{P}}$ and set $\nu = |I|$.

Consider a graph G having a vertex for each interval I_k^l and I_k^r, $k = 1, \ldots, |\bar{\mathcal{P}}|$, and an edge between two vertices if an intersection between two distinct intervals exist, i.e., an edge is related to the presence/absence of mutual compatibility between two distinct reads, then G is a variant of *interval graph* (see Fishburn [1985]), more simply, an *ALU Graph* (ALUG). The class of the ALUGs can be seen as a specialization of the class of the *Interval Graphs* (IGs). In fact, the following proposition holds:

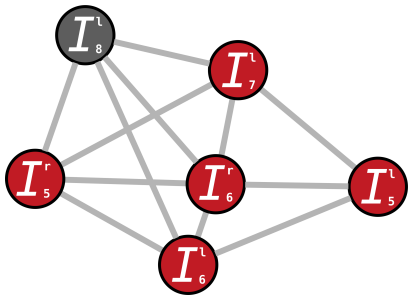
Proposition 4. *The class of the IGs strictly contains the class of the ALUGs.*

Proof. A generic interval $[l_k, r_k]$ of an interval graph is completely characterized by the left and right margins l_k and r_k, respectively. In a similar way, a generic interval of an ALUG A is completely characterized by the interval $[L_k, R_k]$ where $L_k = \texttt{begin}(\pi_k^l)$ and $R_k =$

$\texttt{end}(\pi_k^l) + Y_{max}$ for each left read and $L_k = \texttt{begin}(\pi_k^r) - Y_{max}$ and $R_k = \texttt{end}(\pi_k^r)$ for each right read, respectively. We can transform any ALUG A into an IG G by mapping each interval $[l_k, r_k]$ of G into the interval $[\texttt{begin}(\pi_k^l), \texttt{end}(\pi_k^l) + Y_{max}]$ or $[\texttt{begin}(\pi_k^r) - Y_{max}, \texttt{end}(\pi_k^r)]$ of A (see e.g., Figure 13a). Hence, any ALUG is also an IG. To complete the proof, it is sufficient to show that the converse is not true. In fact, the class of ALUGs is characterized by always having an even number of intervals, which in turn implies that an ALUG does not contain graphs with an odd cardinality of vertices. In contrast, the class of the IGs may also include such graphs. □



(a) Interval representation for the PAIRP



(b) Graph representation for the PAIRP

Figure 13: The ALUG is based on the reads showed in Figure 12b. In particular, Figure 13 shows the mapping of improper reads to a segment of the reference genome, while Figure 13b shows the clique associated to the RCI represented in Figure 13a

Given an instance of the PAIRP and its corresponding ALUG, G, we observe that by definition, the RCIs correspond to *cliques* in G, i.e., to complete subgraphs of G. This is because any RCIs represent a set of improper reads all mutually compatible with ALU insertions. We also recall that a maximal clique of a graph is a clique that is not a subset

of a larger clique and a maximum clique is a clique of maximum size. Then, the following result holds for ALUGs:

Proposition 5. *Let* G *be an ALUG. Then* G *contains at most* v *maximal cliques.*

Proof. The statement follows from observing that the class of the ALUG is contained in the class of interval graphs and that the number of maximal cliques in a interval graph is proven to be at most equal to the number of the vertices (see Fulkerson et al. [1965]) □

The procedure is described in Fulkerson et al. [1965] and consists in deleting a random vertex on the graph and simply list it together with its neighbours in the reduced graph. If the original graph has n vertices, this yields a set of n cliques. The maximal ones of these n cliques are the maximal ones in the graph and there can be at most n. This implies that the maximum clique problem can be solved in polynomial time in an ALUG.

A representation of MIPAIRP by ALUGs consist of considering multiple set of intervals I, each of them coming from a different individual. Hence, let us define the set $\mathcal{I}$ as the union of all I^i where i represent the individual and $G(\mathcal{I})$ is an ALUG for the MIPAIRP. Moreover, it is trivial to see that RCI corresponds to cliques in $G(\mathcal{I})$ and Proposition 5 is valid also for $G(\mathcal{I})$.

In the next section we exploit Proposition 5 to develop an exact approach for the solution of the MIPAIRP based on integer programming.

4.3 THE COMPLEXITY OF PAIRP AND MIPAIRP

Given an instance of the PAIRP/MIPAIRP and its corresponding ALUG, G, we note that in any optimal solution to the instance, a false positive

read will always be a read that does not belong to any RCI selected. Hence, in any optimal solution to the instance, a false positive read corresponds to a clique of the ALUG having cardinality '1'. This insight allows us to consider the PAIRP and the MIPAIRP as instance of the MCPP in a particular class of graphs [Michael et al. 1979]. The MCPP is known to be *NP*-hard in general [Michael et al. 1979]; in this section we will show that the MCPP (i) remains hard even when restricted to the class of the ALUG, (ii) can be solved in polynomial time for the PAIRP version if the cost functions h and g satisfy some specific properties while (iii) the MIPAIRP can not be solved optimally in polynomial time unless *P=NP*.

Before proceeding, we introduce some notation that proves useful throughout the section. We say that a set function f is *zero-cardinal* if $f(\emptyset) = 0$; *non-negative* if f assumes only non-negative values; and *non-decreasing* if $f(T) \leqslant f(S)$ for any $T \subseteq S \subseteq V$. We say that f is sub-modular if it satisfies the following property [Schrijver 2003]:

$$f(S \cup \{u\}) + f(T) \leqslant f(S) + f(T \cup \{u\}) \quad \forall\, T \subseteq S \subseteq V, u \in V \setminus S$$

We say that f is a *polymatroid rank function* if it is zero-cardinal, non-decreasing, non-negative, and sub-modular. Moreover, similarly to [Gijswijt et al. 2007], we define a *value-polymatroid set function* f as a zero-cardinal, non-decreasing, non-negative set function that satisfies the following property:

$$f(S \cup \{u\}) + f(T) \leqslant f(S) + f(T \cup \{u\}) \quad \forall\, S, T \subseteq V : f(S) \geqslant f(T), u \in V \setminus (S \cup T).$$

Note that a value-polymatroid set function is also polymatroidal, but the converse is generally not true [Gijswijt et al. 2007]. Finally, a set function f is size-defined sub-modular if there exists a function $\Phi : [0 \dots |V|] \to \mathbb{R}_0^+$ such that $f(S) = \Phi(|S|)$, for any $S \subseteq V$. As shown in [Gijswijt et al. 2007], a size-defined sub-modular set function f is both value-polymatroidal and polymatroidal. The fact that ALUGs are

variants of the class of interval graphs and the previous definitions turn out to be useful to investigate the complexity of the PAIRP. Specifically, denote $C(G)$ the set of cliques of G and set:

$$f(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ g(C) & \text{if } |C| = 1 \\ h(C) & \text{if } |C| \geqslant 2 \end{cases} \qquad \forall C \in C(G) \tag{16}$$

Then, the following propositions holds:

Proposition 6. *Solving the PAIRP on an ALUG has at least the same complexity of solving the MCPP on an interval graph.*

Proof. Given a generic interval graph G, let define $I(G)$ the set of all the intervals of G. A generic interval $[l_k, r_k]$ of an interval graph is completely characterized by the left and right margins l_k and r_k, respectively. Let define $\Lambda = \bigcup_{k \in I(G)} \{l_k, r_k\}$ as the union of the extreme points of each interval k. Therefore, let define $\lambda = \min_{x,y \in \Lambda} |x - y|$ as the smallest space between intervals. Furthermore, let define $\epsilon = \max_k r_k$ the rightmost extreme point of the rightmost interval. We can now generate the sets of improper read pairs $\bar{\mathcal{P}}'$ using the following transformation. For each interval $k \in I(G)$ we generate an read pair π_k having $\texttt{begin}(\pi_k^l) = l_k$, $\texttt{end}(\pi_k^l) = r_k$, $\texttt{begin}(\pi_k^r) = l_k + 3 * \epsilon$, $\texttt{end}(\pi_k^r) = r_k + 3 * \epsilon$ and $Y_{max} = \lambda$. The ALUG G' induced by $\bar{\mathcal{P}}'$ has $I(G') = \{I^l \cup I^r\}$ as the set of all the "left" and "right" intervals. By construction it is easy to se that G' is the disjoint union of the graph induced by I^l and I^r and that G is isomorphic to both of them. Hence, solving the PAIRP on G' has at least the same complexity of solving the MCPP on an interval graph. □

Proposition 7. *The decision version of the PAIRP is NP-complete even when the cost function f is restricted to polymatroidal set functions.*

Proof. The statement follows Proposition 6 and that the MCPP on an interval graph is *NP*-complete when the cost function is polymatroidal [Gijswijt et al. 2007] □

In general, it is easy to realize that the decision version of the PAIRP is $\mathcal{NP}$-complete for any cost function $f(C) = \psi(|C|)\sigma(C)$ such that $\psi : [0...|V|] \to R + 0$ and $\sigma(C)$ is a generic function on C. In fact, such a case also includes the *rooted-TSP cost function on a tree* (see Gijswijt et al. [2007]) which is trivially polymatroidal. Although Proposition 7 states that the decision version of the PAIRP is in general $\mathcal{NP}$-complete, it is worth noting that in some special cases the problem can be still solved in polynomial time. For example, the following proposition holds:

Proposition 8. *Let $G = (V, E)$ be an ALUG and $f : C(G) \to \mathbb{R}_0^+$ a value-polymatroidal cost function. If G is also an interval graph then it is possible to compute a minimum cost partition into cliques of G in polynomial time.*

Proof. The statement follows from the fact that the minimum clique partition problem on an interval graph can be solved in polynomial time when the cost function is a value-polymatroidal set function [Gijswijt et al. 2007]. □

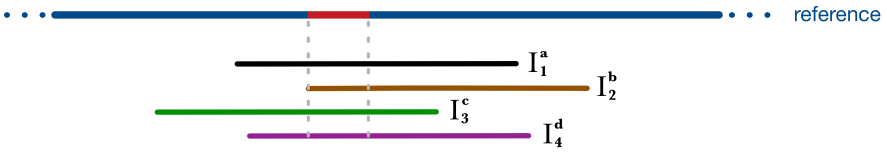
Proposition 8 turns out useful to show that if G is an interval graph the PAIRP can be solved in polynomial-time when the following objective function is used:

$$f_\alpha(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ c1 & \text{if } |C| = 1 \\ c2 & \text{if } |C| \geqslant 2 \end{cases} \qquad \forall C \in C(G) \tag{17}$$

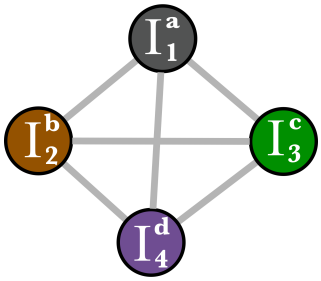
where c_1 and c_2 are two constants such that $0 < c_1 \leqslant \alpha c_1 < c_2 \leqslant (\alpha + 1)c_1$, and α is a positive integer such that $2 \leqslant \alpha \leqslant |V| - 1$. In fact, in such a case it is easy to see that the set function $f_\alpha(C)$ is size-defined submodular, hence value-polymatroidal; thus, if $G(I)$ is an ALUG, by Proposition 8 the PAIRP can be solved in polynomial time. Moreover, it is worth noting that the optimal solution to the problem can be characterized when considering function (17). In fact, the following proposition holds:

Proposition 9. *Consider a graph* $G = (V, E)$ *and a cost function* f_α *defined as in (17). Then, there exists a minimum cost partition into cliques of* G*, say* $P^\star$*, such that none of the cliques in* $P^\star$ *have cardinality greater than* 1 *and smaller than* $\alpha + 1$*. Moreover, if* $P^\star$ *contains cliques of cardinality greater than or equal to* $\alpha + 1$ *then at least one of them is a maximal clique of* G*.*

Proof. By contradiction, suppose there exists a clique $C \in P^\star$ such that $2 \leqslant |C| \leqslant \alpha$. Then, due to the nature of f_α, it is possible to obtain a lower cost partition into cliques of G by just breaking C into $|C|$ cliques of cardinality 1. In fact, in such a case we would have that $\sum_{v \in C} f_\alpha(\{v\}) = |C|c_1 < c_2 = f_\alpha(C) \leqslant (\alpha + 1)c_1$. However, this contradicts the claim that $P^\star$ has minimum cost. Hence, $P^\star$ does not contain cliques having cardinality between 2 and α. Now, assume that $P^\star$ contains a clique $C \in P^\star$ such that $|C| \geqslant \alpha + 1$. Since f_α is non-decreasing we have that $f_\alpha(C) \geqslant f_\alpha(T)$,for any $T \in P^\star$. If C is not a maximal clique of G then there exists some $t \in V \setminus C$ such



(a) Interval representation for the MIPAIRP



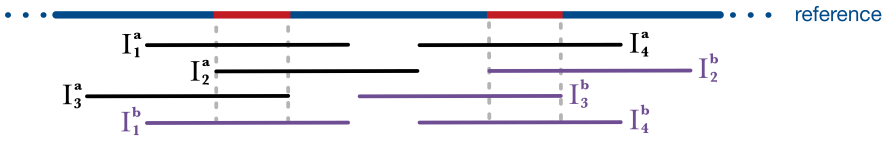
(b) Graph for the MIPAIRP

Figure 14: Example of ALUG for MIPAIRP. Figure 15a shows the mapping of improper reads coming from multiple individuals, identified by different colours, to a segment of the reference genome, while Figure 14b shows the cliques associated to the RCIs represented in Figure 14a

that $C \cup \{t\}$ is a clique in G. Note that t belongs to some $T \in P^\star \setminus C$ and that since f_α is non-decreasing, it holds that $f_\alpha(C) \geqslant f_\alpha(T) \geqslant f(T \setminus \{t\})$. Observe also that since f_α is α-value-polymatroidal, it holds that $f_\alpha(C \cup \{t\}) + f_\alpha(T \setminus \{t\}) \leqslant f_\alpha(C) + f_\alpha(T)$. Hence, it is possible to enlarge C until it becomes a maximal clique of G without getting worse than the cost of $P^\star$. □

Based on Propositions 7,8 and 9 a recursive dynamic programming approach for the PAIRP is provided in Gijswijt et al. [2007].

On the other hand, in the multiple individual scenario, the presence of multiple individuals need to be taken in account in the definition of the objective function for the MIPAIRP. For example, by penalising all cases in which an RCI is identified only by improper reads belonging all to different individuals. Since an RCI in an ALUG is represented by a clique, these biased RCI may arise whenever the cardinality of the clique is equal to the number of individuals contributing in the same clique (see Figure 14).



(a) Interval representation for the MIPAIRP

(b) Graph representation for the MIPAIRP

Figure 15: Example of ALUG for MIPAIRP. Figure 15a shows the mapping of improper reads coming from multiple individuals to a segment of the reference genome, while Figure 15b shows the cliques associated to the RCIs represented in Figure 15a

Specifically, cliques having nodes coming from a single individual are more likely to represent RCI than a cliques having each node coming from a different individual (see Figure 15) and we can state the following proposition:

Proposition 10. *Given an ALUG* G *containing intervals coming from multiple individuals, each clique having all the nodes belonging to a different individual does not represent a RCI for any of the individuals.*

Proof. The statement follows by observing that a RCI is present in an individual i, iff the ALUG $G_i = (V_i, E(V_i))$ contains a clique C of cardinality bigger or equal to 2. This is impossible if for all the cliques C in the ALUG $G = (V, E(V))$ where the cardinality of the clique C is equal to the number of individuals belonging to that clique. □

Based on Proposition 10, the new following cost function seems much more well-suited:

$$f_\alpha(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ c1 & \text{if } |C| = 1 \\ c2 + c3 \cdot \omega_C & \text{if } |C| \geqslant 2 \end{cases} \qquad \forall C \in C(G) \tag{18}$$

where $c1$, $c2$ and $c3$ are three constants such that $0 < c_1 \leqslant \alpha c_1 < c_2 \leqslant c_3 \leqslant (\alpha+1)c_1$, α is a positive integer such that $2 \leqslant \alpha < |V| - 1$, and ω_C the number of individuals contained in the clique $C \in C(G)$. Unfortunately, in such a case it is easy to see in the following example that the new set function (18) is not size-defined submodular, hence value-polymatroidal, but a simple submodular rank function.

Given the graph G in Figure 16, $S = \{a, b, c\}$, $T = \{e, f, g\}$, $u = \{d\}$ and $c_1 = 1$ and $c_2 = c_3 = 2$. The main property of the polymatroid set function for function (18) is not satisfied because:

$$f(S \cup \{u\}) + f(T) \leqslant f(S) + f(T \cup \{u\}) \quad \forall\, S, T \subseteq V : f(S) \geqslant f(T), u \in V \setminus (S \cup T).$$

gives as result that:

$$2c_2 + 3c_3 \leqslant 2c_2 + 2c_3$$

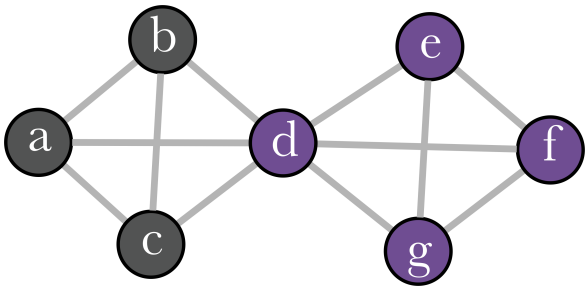


Figure 16: Example of a graph where function(18) prove to be not value-polymatroid. This figure shows a graph composed of vertices belonging to two different individuals X and Y. In particular $X = \{a, b, c\}$ and $Y = \{d, e, f, g\}$.

Based on Propositions 10 and cost function (18) the recursive dynamic programming approach provided in Gijswijt et al. [2007] can not be used for the MIPAIRP.

4.4 AN INTEGER PROGRAMMING MODEL FOR THE MIPAIRP

The *NP*-hardness of the MIPAIRP justifies the development of exact and approximate solution approaches for the problem. In this section we present an integer programming model for the MIPAIRP. To this end, given a vertex $v \in V$, we denote $C_v = \{C \in \mathcal{C}(G) : v \in C\}$. Moreover, denote y_C as a decision variable equal to 1 if the clique $C \in \mathcal{C}(G)$ is selected in the optimal solution to the problem and 0 otherwise. Then, Formulation 4 is a valid formulation for the MIPAIRP:

Formulation 4.

$$\min \sum_{C \in \mathcal{C}(G)} f(C) y_C \tag{19a}$$

$$\text{s.t.} \sum_{C \in \mathcal{C}_v(G)} y_C = 1 \quad \forall v \in V \tag{19b}$$

$$y_C \in \{0, 1\} \quad \forall C \in \mathcal{C}(G) \tag{19c}$$

Constraints (19b) impose that each vertex $v \in V$ belongs to a selected clique $C \in C(G)$ and constraints (19c) impose the integrality

on variables y_C. Formulation 4 is characterized by an exponential number of variables and constraints and its linear relaxation can be exactly solved by using column generation techniques. Specifically, observe that a variable with negative reduced cost in the linear relaxation of Formulation 4 corresponds to a dual constraint violated by the current dual solution. Denoted μ_v as the dual variables associated with constraints (19b), the dual of the linear relaxation of Formulation 4, denoted by LP1, is characterized by the following constraints:

$$\sum_{v \in V : v \in C} \mu_v \leqslant f(C) \quad \forall C \in \mathcal{C}(G) \tag{20}$$

Constraints (20) are violated if there exists a clique $C \in C(G)$ such that $\sum_{v \in V : v \in C} \mu_v > f(C) \quad \forall C \in \mathcal{C}(G)$. The existence of such a clique can be checked in polynomial time by using Proposition 5 and this in turn implies that the linear relaxation of LP1 can be solved in polynomial time. Interestingly, if the cost function f is defined as in (18) then Formulation 4 can be rewritten as follows. Denote x_v as a decision variable equal to 1 if vertex $v \in V$ forms a clique of cardinality 1 in the optimal solution to the problem and 0 otherwise. Moreover, denote $\hat{\mathcal{C}}(G)$ as the set of all maximal cliques in G having cardinality greater or equal to 2 and $\hat{\mathcal{C}}(G) = \{C \in \mathcal{C}(G) : v \in C, |C| \geqslant 2\}$.

Then, Formulation 5 is a valid formulation for the MIPAIRP.

Formulation 5.

$$\min \sum_{v \in V} c_1 x_v + \sum_{C \in \hat{\mathcal{C}}(G)} (c_2 + c_3 \omega_C) y_C \tag{21a}$$

$$\text{s.t.} \sum_{C \in \hat{\mathcal{C}}(G)} y_C + x_v \geqslant 1 \quad \forall v \in V \tag{21b}$$

$$x_v \in \{0, 1\} \quad \forall v \in V \tag{21c}$$

$$y_C \in \{0, 1\} \quad \forall C \in \hat{\mathcal{C}}(G) \tag{21d}$$

Formulation 5 has the benefit of being polynomial-sized, due to Proposition 5, hence in principle its relaxation does not require the

use of column generation techniques to be solved.

Formulations 4 and 5 can be strengthened by adapting appropriately the inequalities described in Grötschel et al. [1990],Bandelt et al. [1999] and Oosten et al. [2001]. Moreover, additional valid inequalities can be considered. Specifically, given a pair of distinct vertices $v, w \in V$, we say that v *dominates* w if (i)$N(w) \subset N(v)$ and (ii) for any $C_v \in \hat{\mathcal{C}}(G)$ and $C_w \in \hat{\mathcal{C}}(G)$ it holds that $|C_v| > |C_w|$. In such a case, we also say that v is *dominating* and w is *dominated*. Dominated vertices are irrelevant to the clique partitioning of G, since in any optimal solution to the problem they will always be identified as cliques of cardinality one. Hence, both formulations can be strengthened by adding the following valid inequalities, whose proof trivially follows from Proposition 8 and the definition of domination:

Proposition 11. *Let v be a dominating vertex in V. Then, the inequality*

$$\sum_{C_v \in \hat{\mathcal{C}}(G)} y_C \geqslant 1$$

is valid for both Formulations 4 and 5.

4.5 EXPERIMENTAL RESULTS

In this section we analyze the performance of our model in solving instances of the *Multiple Individuals Polymorphic ALU Insertions Recognition Problem* (MIPAIRP). Our experiments were motivated mainly to measure the runtime performances of our model in tackling real size instances of the MIPAIRP. We emphasize that our experiments simply aim to evaluate the computational performance of our model; we neither attempt to study the efficiency of it to predict ALU insertions across the genomes of a population of individuals nor to compare its accuracy versus ALU insertions predictors using an objective function that is different from the one used in Sveinbjörnsson et al. [2012].

Datasets	Reads per individuals	Number of individuals	Max Cliques (Avg.)
Set-1$_\alpha$	50	1 000	23 508.6
Set-2$_\alpha$	100	1 000	40 656.9
Set-3$_\alpha$	150	1 000	50 484.7
Set-1$_\beta$	50	2 000	40 656.0
Set-2$_\beta$	100	2 000	55 345.7
Set-3$_\beta$	150	2 000	55 618.8
Set-1$_\gamma$	50	3 000	50 456.3
Set-2$_\gamma$	100	3 000	58 577.3
Set-3$_\gamma$	150	3 000	59 298.3

Table 15: Statistics of Set-x$_\alpha$, Set-x$_\beta$, Set-x$_\gamma$

4.5.1 *Implementation*

We modelled the MIPAIRP with PYOMO [Hart et al. 2017], a open-source software package that supports a diverse set of optimization capabilities for formulating, solving, and analyzing optimization models, using Gurobi Optimizer version 6.5.2 build v6.5.2rc1. The experiments were performed on an Intel Core i7-4930K CPU, 3.40GHz, equipped with 64 GByte RAM and operating system Ubuntu release 16.04.3 LTS (kernel GNU/Linux 4.4.0-51-generic).

4.5.2 *Benchmark Instances*

We tested our model on three datasets, hereafter denoted as "Set-1", "Set-2" and "Set-3", characterized by having 50 000, 100 000, 150 000, intervals on the real segment [1, 60 000], respectively, each having

length 120 and insertion distance Y_{max} fixed to 200. Hence, we also considered increasing size instances of the MIPAIRP obtained by augmenting the number of individuals from each "set-x", $x \in \{1,3\}$, by $\alpha = 1\,000$, $\beta = 2\,000$ and $\gamma = 3\,000$, respectively. We generate for each "set-x", $x \in \{1,3\}$, 15 random instances of the MIPAIRP for a total of 135, that are hereafter grouped into three datasets of 45 denoted as "Set-x_α", "Set-x_β" and "Set-x_γ", $x \in \{1,3\}$, respectively. Table 15 shows the main characteristics of the 135 instances. In particular, the table is composed by three main sections, each one corresponding to the datasets "Set-x_α", "Set-x_β" and "Set-x_γ", $x \in \{1,3\}$, previously described.

The first column of Table 15 reports the name of the datasets. The second column reports the number of reads generated for each individual. The third column reports the overall number of individuals analyzed in each given instance. Finally, the fourth column reports the average number of maximal cliques contained in all the 15 instances.

4.5.3 *Computational performances*

Table 16 shows the performances of Formultaion 5 in solving the 135 instances described is Section 4.5.2. In particular, the table is constituted by three main sections, each one corresponding to the datasets "Set-x_α", "Set-x_β" and "Set-x_γ", $x \in \{1,3\}$, previously described. The first column of Table 16 reports the name of the datasets. The second column reports the average number of constraints generated for each instance. The third column reports the average number of variables generated for each instance. The fourth column reports the average number of iteration of the dual simplex executed by the solver. Finally, the fifth column reports the average time spent by the solver finding

Datasets	# of const. (Avg.)	# of var. (Avg.)	# of it. (Avg.)	Sol. Time (Avg.)
Set-1_α	50 001	73 509.6	49 732.3	5.63558
Set-2_α	100 001	140 658.0	93 724.2	26.6198
Set-3_α	150 001	200 486.0	135 418.0	104.23
Set-1_β	100 001	140 657.0	98 201.0	21.2821
Set-2_β	200 001	255 347.0	186 361.0	96.5079
Set-3_β	300 001	262 243.0	191 627.0	95.1027
Set-1_γ	150 001	200 457.0	146 110.0	42.974
Set-2_γ	300 001	358 578.0	276 362.0	187.612
Set-3_γ	450 001	509 299	389 056.0	459.879

Table 16: Performances of Formulation 5 on Set-x_α , Set-x_β , Set-x_γ

the optimal solution for each instance. As a general trend, Table 16 shows that our model is able to solve all the instances to optimality in less than 3 minutes. Only the set of instances "Set-3_γ" needs on average more than 7 minutes to be solved. Moreover, the table shows a solving time four times greater when passing from 1 000 to 2 000 individuals (apart for Set-3_α/Set-3_β) and a solving time two times greater when going from 2 000 to 3 000 individuals.

5

CONCLUSION

In this chapter we highlight the main contributions of our research and summarize the main results. Then, we present some potential avenues for future research.

5.1 CONTRIBUTION

In this dissertation we have studied problem arising it the domain of *Genome-Wide Association Studies* (GWAS) and more in general Computational Biology. GWAS have evolved over the last ten years into a powerful tool for investigating the genetic architecture of human disease. Armed with the information from GWAS scientists got a better understanding of how diseases develop and how they might be diagnosed and treated. Over the last few years, GWAS in humans have revealed many genes associated with disease and provided an insight into the mechanisms of a number of complex diseases. This dissertation has provided a systematic analysis of the mathematical models used to describe three *NP*-hard problems arising in the domain of *Genome-Wide Association Studies* (GWAS). The contributions and the observations made in our work helped illustrate the power of *Operations Research* approaches for some problems arising in the domain of GWAS, and some of them characterized by complex sets of constraints.

In particular, in Chapter 2 we have investigated, for the first time, a recent version of PPH, called the *Pure Parsimony Haplotyping under Uncertain Data* (PPH-UD). This version mainly arises when the input

genotypes are not accurate, i.e., when some single nucleotide polymorphisms are missing or affected by errors. We proposed an exact approach to solution of PPH-UD based on an extend version of Catanzaro et al. [2010b] class representative model for PPH, possibly one of the best integer programming models described so far in the literature on PPH. We proposed a number of reduction rules to improve the performances of the model and showed how to embody the perfect phylogeny constrains in order to represent known familial relationships among members of the input population.

Afterwards, in Chapter 3 we investigated the *Parsimonious Loss of Heterozygosity Problem* (PLOHP), a *NP*-hard combinatorial optimization problem consisting of finding the minimum cost clique cover problem on a MPTG. The optimal solution to the PLOHP has a remarkable importance in practice, as it enables the association of major human diseases with chromosomic regions from patients that exhibit loss of heterozigosity events. Practical instances of the PLOHP are usually characterized by a very large size and this fact usually prevents the use of exact solution approaches such as those described in Catanzaro et al. [2013] and Halldórsson et al. [2011b]. Moreover, we investigated ways to speed up the best known exact solution algorithm for the PLOHP as well as techniques to enlarge the size of the instances that can be optimally solved. In particular, we presented a *Branch-and-Price* (B&P) algorithm for the PLOHP and we developed a number of preprocessing techniques and decomposition strategies to dramatically reduce the size of its instances. Computational experiments showed that the proposed algorithm is 10-30x faster than previous approaches described in the literature. The new algorithm enables the solution of very large practical instances of the PLOHP, containing over 6 000 trios and SNPs.

Finally, in Chapter 4 we investigate the *Multiple Individuals Polymorphic ALU Insertions Recognition Problem* (MIPAIRP), i.e., the problem of partitioning suspected polymorphisms of a set of individuals into the minimum number of deletion areas. Specifically, we showed that the MIPAIRP can be formulate as particular instance of the clique partition problem in a *undirected interval graph* G, and we prove the general *NP*-hardness of the problem. Moreover, we extended the results described in Sveinbjörnsson et al. [2012] by providing a state-of-the-art integer programming formulation and a possible strengthening valid inequalities able to exactly solve real instances of the MIPAIRP containing up to 3 000 individuals and 3 000 SNPs within 12 hour computing time. The overall performances could be further enhanced by implementing the preprocessing techniques presented in Chapter 3.

5.2 PERSPECTIVES

The primary goals of this thesis have been the study of problems arising in the field of biological systems under the OR point of view. Our contributions and the observations made in our work helped illustrate how profitably the sophisticated optimization techniques of OR can be applied to a non-mathematical domain such as GWAS. Moreover, our work also pose a number of interesting open questions for the specific research issues attacked in this thesis and for the research in the field of GWAS, in general.

In particular, in Chapter 2 we have investigated the *Pure Parsimony Haplotyping under Uncertain Data* (PPH-UD) which take in account input genotypes that are not accurate. The lack of algorithm able to take in account bias and errors is one of the well known challenge in the analysis of biological data. In fact, several procedures have been developed for dealing with errors in linkage studies, forensic analy-

ses and non-invasive genotyping before using predictive approaches Paolella et al. [1983]. Our exact approach to solution of PPH-UD based on an extend version of Catanzaro et al. [2010b] proposed in Chapter 2 should encourage the use of mathematical programming in solving these kind of problems. More so if such possible extensions have been carried out without compromising the performances in solving even larger instances. Furthermore, since the introduction in the last few years of error rate monitoring in the DNA sequencing phase [Manley et al. 2016], the use of a non-binary error mask matrix for the PLOHP deserve for sure future research efforts.

Afterwards, in Chapter 3 we investigated the *Parsimonious Loss of Heterozygosity Problem* (PLOHP), a *NP*-hard combinatorial optimization problem consisting of finding the minimum cost clique cover problem on a MPTG. The decomposition methods presented allowed us to solve 10-30x faster than previous approaches described in the literature and very large practical instances of the PLOHP, containing over 6 000 trios and SNPs. Another avenue of improvement involves analysing instances that could be solved faster by using a polynomial algorithm. This can not be done in general since PLOHP is *NP*-hard, but if we are able in polynomial time to characterise a specific instance as an interval graph, instead of a MPTG, the we could use the polynomial algorithm proposed by Gijswijt et al. [2007]. Hopefully, future research efforts will provide new insights on the combinatorics of MPTG and will enable the analysis of even larger genomic fragments.

Moreover, in Chapter 4 we investigate the *Multiple Individuals Polymorphic ALU Insertions Recognition Problem* (MIPAIRP), a *NP*-hard combinatorial optimization problem consisting of partitioning suspected polymorphisms of a set of individuals into the minimum number of deletion areas. The IP formulation presented allowed to to exactly

solve real instances of the MIPAIRP containing up to 3 000 individuals and 3 000 SNPs within 12 hour computing time. Our results give perspective on the development of future approaches to solution of the problem that may turn out to be useful in practical applications.

Finally, even if all the problems faced in this thesis are *NP*-hard, we showed how to make some progress and be able to efficiently solve large scale instances by providing compact formulations and efficient decomposition methods. Moreover, we have made improvements in being able to efficiently solve large scale instances for well known problems in the domain. It is our belief that the proposed methods and solving techniques will be helpful in analysing real-size datasets to identify new variation sites associated with genetic disease, and the basis for the development of new mathematical models for the arising problems in the GWAS domain.

BIBLIOGRAPHY

Achterberg, T. 2009. "SCIP: Solving constraint integer programs." http://mpc.zib.de/index.php/MPC/article/view/4, *Mathematical Programming Computation* 1 (1): 1–41.

Albert, V. A. 2005. *Parsimony, phylogeny, and genomics*. Oxford Bioscience, UK.

Alexander, R. 1996. *Optima for Animals*. Princeton paperbacks. Princeton University Press.

Altshuler, D., J. N. Hirschhorn, M. Klannemark, C. M. Lindgren, M. C. Vohl, J. Nemesh, C. R. Lane, S. F. Schaffner, S. Bolk, C. Brewer, T. Tuomi, D. Gaudet, T. J. Hudson, M. Daly, L. Groop, and E. S. Lander. 2000. "The common PPARgamma Pro12Ala polymorphism is associated with decreased risk of type 2 diabetes." *Nat Genet* (Whitehead Institute/MIT Center for Genome Research, Cambridge, Massachusetts, USA.) 26, no. 1 (): 76–80.

Andrade, A. de, M. Wang, M. F. Bonaldo, H. Xie, and M. B. Soares. 2011. "Genetic and epigenetic variations contributed by Alu retrotransposition." *BMC Genomics* 12 (1): 617.

Aringhieri, R., and D. Catanzaro. 2011. "Optimal solutions for the balanced minimum evolution problem." *Computers and Operations Research* 38:1845–1854.

Bafna, V., D. Gusfield, S. Hannenhalli, and S. Yooseph. 2004. "A note on efficient computation of haplotypes via perfect phylogeny." *Journal of Computational Biology* 11 (5): 858–866.

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

Bandelt, H.-J., M. Oosten, J. H. G. C. Rutten, and F. C. R. Spieksma. 1999. "Lifting theorems and facet characterization for a class of clique partitioning inequalities." *Operations Research Letters* 24 (5): 235–243.

Banga, J. R. 2008. "Optimization in computational systems biology." *BMC Syst Biol* (Instituto de Investigaciones Marinas, CSIC (Spanish Council for Scientific Research), C/Eduardo Cabello 6, 36208 Vigo, Spain. julio@iim.csic.es) 2 (): 47.

Bell, G. I., S. Horita, and J. H. Karam. 1984. "A polymorphic locus near the human insulin gene is associated with insulin-dependent diabetes mellitus." *Diabetes* 33, no. 2 (): 176–183.

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

Błażewicz, J., P. Formanowicz, and M. Kasprzak. 2005. "Selected combinatorial problems of computational biology." *European Journal of Operational Research* 161 (3): 585–597.

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

Booth, K. S., and G. S. Lueker. 1976. "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms." *Journal of Computer and System Sciences* 13 (3): 335–379.

Brown, D., and I. M. Harrower. 2004. "A New Integer Programming Formulation for the Pure Parsimony Problem in Haplotype Analysis." In *Proceedings of the Fourth Annual Workshop Algorithms in Bioinformatics*, ed. by I. Jonassen and J. Kim, 3240:254–265. Springer-Verlag.

— . 2006. "Integer Programming Approaches to Haplotype Inference by Pure Parsimony." *IEEE Transactions, Computational Biology and Bioinformatics* 3 (2): 141–154.

Cargill, M., D. Altshuler, J. Ireland, P. Sklar, K. Ardlie, N. Patil, N. Shaw, C. R. Lane, E. P. Lim, N. Kalyanaraman, J. Nemesh, L. Ziaugra, L. Friedland, A. Rolfe, J. Warrington, R. Lipshutz, G. Q. Daley, and E. S. Lander. 1999. "Characterization of single-nucleotide polymorphisms in coding regions of human genes." *Nat Genet* (Whitehead Institute/MIT Center for Genome Research, Cambridge, Massachusetts 02139, USA. lander@genome.wi.mit.edu) 22, no. 3 (): 231–238.

Catanzaro, D. 2009. "The minimum evolution problem: Overview and classification." *Networks* 53 (2): 112–125.

— . 2011. "Estimating Phylogenies from Molecular Data." In *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, ed. by R. Bruni, 149–176. Springer New York.

Catanzaro, D., S. Chaplick, S. Felsner, B. V. Halldórsson, M. M. Halldórsson, T. Hixon, and J. Stacho. 2014. "Max-Point-Tolerance Graphs." *Discrete Applied Mathematics*, no. Submitted. Tech. Rep. TR-2014-02 available at G.O.M. - Computer Science Department - Université Libre de Bruxelles.

Catanzaro, D., M. Labbé, and B. V. Halldórsson. 2013. "An integer programming formulation of the parsimonious loss of heterozygosity problem." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 10 (6): 1391–1402.

Catanzaro, D., M. Andrien, M. Labbe, and M. Toungouz-Nevessignsky. 2010a. "Computer-aided human leukocyte antigen association studies: a case study for psoriasis and severe alopecia areata." *Hum Immunol* (Graphs)(Mathematical Optimization, Computer Science Department, Universite Libre de Bruxelles Brussels, Belgium. dacatanz@ulb.ac.be) 71, no. 8 (): 783–788.

Catanzaro, D., A. Godi, and M. Labbé. 2010b. "A Class Representative Model for Pure Parsimony Haplotyping." *INFORMS J. on Computing* 22:195–209.

Catanzaro, D., and M. Labbé. 2009. "The pure parsimony haplotyping problem: overview and computational advances." *International Transactions in Operational Research* 16 (5): 561–584.

Chakravarti, A. 1998. "It's raining SNPs, hallelujah?" *Nature genetics* 19 (3): 216–217.

Chen, K., J. W. Wallis, M. D. McLellan, D. E. Larson, J. M. Kalicki, C. S. Pohl, S. D. McGrath, M. C. Wendl, Q. Zhang, D. P. Locke, X. Shi, R. S. Fulton, T. J. Ley, R. K. Wilson, L. Ding, and E. R. Mardis. 2009. "BreakDancer: an algorithm for high-resolution mapping of genomic structural variation." *Nat Methods* (The Genome Center, Washington University School of Medicine, St. Louis, Missouri, USA. kchen22@wustl.edu) 6, no. 9 (): 677–681.

Clark, V. J., N. Metheny, M. Dean, and R. J. Peterson. 2001. "Statistical estimation and pedigree analysis of CCR2-CCR5 haplotypes." *Hum Genet* (Laboratory of Genomic Diversity, NCI at Frederick, MD 21702, USA. clarkv@ncifcrf.gov) 108, no. 6 (): 484–493.

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

Clausen, J., and M. Perregaard. 1999. "On the best search strategy in parallel branch-and-bound:Best-First Search versus Lazy Depth-First Search." *Annals of Operations Research* 90 (0): 1–17.

Conforti, M., G. Cornuéjols, and G. Zambelli. 2014. *Integer Programming*. Graduate Texts in Mathematics. Springer International Publishing.

Conrad, D. F., D. Pinto, R. Redon, L. Feuk, O. Gokcumen, Y. Zhang, J. Aerts, T. D. Andrews, C. Barnes, P. Campbell, T. Fitzgerald, M. Hu, C. H. Ihm, K. Kristiansson, D. G. MacArthur, J. R. MacDonald, I. Onyiah, A. W. C. Pang, S. Robson, K. Stirrups, A. Valsesia, K. Walter, J. Wei, C. Tyler-Smith, N. P. Carter, C. Lee, S. W. Scherer, and M. E. Hurles. 2010. "Origins and functional impact of copy number variation in the human genome." *Nature* 464, no. 7289 (): 704–712.

Conrad, D. F., T. D. Andrews, N. P. Carter, M. E. Hurles, and J. K. Pritchard. 2006. "A high-resolution survey of deletion polymorphism in the human genome." *Nature Genetics* 38 (1): 75–81.

Consortium, T. I. H. 2003. "The international hapmap project." *Nature* 426 (18): 789–796.

— . 2005. "A haplotype map of the human genome." *Nature* 437 (27): 1299–1314.

Cordaux, R., and M. A. Batzer. 2009. "The impact of retrotransposons on human genome evolution." *Nat Rev Genet* (CNRS UMR 6556 Ecologie, Evolution, Symbiose, Universite de Poitiers, 40 Avenue du Recteur Pineau, Poitiers, France.) 10, no. 10 (): 691–703.

Dahlbäck, B. 1997. "Resistance to activated protein C caused by the factor V R506Q mutation is a common risk factor for venous thrombosis." *Journal of Thrombosis and Haemostasis* 78:483–488.

Deeb, S. S., and *et al.* 1998. "A Pro12Ala substitution in PPAR γ 2 associated with decreased receptor activity, lower body mass index and improved insulin sensitivity." *Nature Genetics* 20:284–287.

Dorman, J. S., R. E. LaPorte, R. A. Stone, and M. Trucco. 1990. "Worldwide differences in the incidence of type I diabetes are associated with amino acid variation at position 57 of the HLA-DQ beta chain." *Proc Natl Acad Sci U S A* (Department of Epidemiol-

ogy, Graduate School of Public Health, University of Pittsburgh, School of Medicine, PA 15261.) 87, no. 19 (): 7370–7374.

Doyle, F. J. 3., and J. Stelling. 2006. "Systems interface biology." *J R Soc Interface* (Department of Chemical Engineering, University of California, Santa Barbara, CA 93106, USA. frank.doyle@icb.ucsb.edu) 3, no. 10 (): 603–616.

Elder, D. A., K. Kaiser-Rogers, A. S. Aylsworth, and A. S. Calikoglu. 2001. "Type I diabetes mellitus in a patient with chromosome 22q11.2 deletion syndrome." *Am J Med Genet* (Division of Endocrinology, Department of Pediatrics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599,USA.) 101, no. 1 (): 17–19.

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

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

Fallin, D., and N. J. Schork. 2000. "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.

Fay, M. P., and M. A. Proschan. 2010. "Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules." *Statistics Surveys* 4:1–39.

Fishburn, P. C. 1985. *Interval Orders and Interval Graphs:A Study of Partially Ordered Sets*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, New York.

Frazer, K. A., et al. 2007. "A second generation human haplotype map of over 3.1 million SNPs." *Nature* (The Scripps Research In-

stitute, 10550 North Torrey Pines Road MEM275, La Jolla, California 92037, USA.) 449, no. 7164 (): 851–861.

Fulkerson, D. R., and O. Gross. 1965. "Incidence matrices and interval graphs." *Pacific Journal of Mathematics* 15, no. 3 (): 835–855.

Fullwood, M. J., C.-L. Wei, E. T. Liu, and Y. Ruan. 2009. "Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses." *Genome Res* (Genome Institute of Singapore, Agency for Science, Technology)(Research, Singapore 138672, Singapore.) 19, no. 4 (): 521–532.

Gijswijt, D., V. Jost, and M. Queyranne. 2007. *Clique partitioning of interval graphs with submodular costs on the cliques*. Les Ulis, FRANCE.

Goedert, M., and M. G. Spillantini. 2006. "A century of Alzheimer's disease." *Science* (Laboratory of Molecular Biology, Medical Research Council, Cambridge CB2 2QH, UK. mg@mrc-lmb.cam.ac.uk) 314, no. 5800 (): 777–781.

Greenberg, H. J., W. E. Hart, and G. Lancia. 2004. "Opportunities for Combinatorial Optimization in Computational Biology." *INFORMS Journal on Computing* 16 (3): 211–231.

Gretarsdottir, S., G. Thorleifsson, S. T. Reynisdottir, A. Manolescu, S. Jonsdottir, T. Jonsdottir, T. Gudmundsdottir, S. M. Bjarnadottir, O. B. Einarsson, H. M. Gudjonsdottir, M. Hawkins, G. Gudmundsson, H. Gudmundsdottir, H. Andrason, A. S. Gudmundsdottir, M. Sigurdardottir, T. T. Chou, J. Nahmias, S. Goss, S. Sveinbjornsdottir, E. M. Valdimarsson, F. Jakobsson, U. Agnarsson, V. Gudnason, G. Thorgeirsson, J. Fingerle, M. Gurney, D. Gudbjartsson, M. L. Frigge, A. Kong, K. Stefansson, and J. R. Gulcher. 2003. "The gene encoding phosphodiesterase 4D confers risk of ischemic stroke." *Nat Genet* (deCODE Genetics, Sturlugata 8, IS-101 Reykjavik, Iceland. solveig.gretarsdottir@decode.is) 35, no. 2 (): 131–138.

Grötschel, M., and Y. Wakabayashi. 1990. "Facets of the clique partitioning polytope." *Mathematical Programming* 47 (1): 367–387.

Gudbjartsson, D. F., G. B. Walters, G. Thorleifsson, H. Stefansson, B. V. Halldorsson, P. Zusmanovich, P. Sulem, S. Thorlacius, A. Gylfason, S. Steinberg, A. Helgadottir, A. Ingason, V. Steinthorsdottir, E. J. Olafsdottir, G. H. Olafsdottir, T. Jonsson, K. Borch-Johnsen, T. Hansen, G. Andersen, T. Jorgensen, O. Pedersen, K. K. Aben, J. A. Witjes, D. W. Swinkels, M. den Heijer, B. Franke, A. L. M. Verbeek, D. M. Becker, L. R. Yanek, L. C. Becker, L. Tryggvadottir, T. Rafnar, J. Gulcher, L. A. Kiemeney, A. Kong, U. Thorsteinsdottir, and K. Stefansson. 2008. "Many sequence variants affecting diversity of adult human height." *Nat Genet* (deCODE Genetics, 101 Reykjavik, Iceland. daniel.gudbjartsson@decode.is) 40, no. 5 (): 609–615.

Gusfield, D. 2001. "Inference of haplotypes from samples of diploid populations: Complexity and algorithms." *Journal of Computational Biology* 8:305–324.

— . 2003. "Haplotype inference by pure parsimony." In *Annual Symposium in Combinatorial Pattern Matching*, ed. by L. N. in Computer Science, 2676:144–155. Springer-Verlag, Berlin, Germany.

Halldórsson, B. V., D. Aguiar, R. Tarpine, and S. Istrail. 2011a. "The Clark phaseable sample size problem: Long-range phasing and loss of heterozygosity in GWASlark phaseable sample size problem: long-range phasing and loss of heterozygosity in GWAS." *Journal of Computational Biology* 18 (3): 323–333.

Halldórsson, B. V., V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. 2003. "Combinatorial Problems Arising in SNP and Haplotype Analysis." In *Discrete Mathematics and Theoretical Computer Science: 4th International Conference, DMTCS 2003 Dijon, France, July 7–12, 2003 Proceedings*, ed. by C. S. Calude, M. J. Dinneen,

and V. Vajnovszki, 26–47. Berlin, Heidelberg: Springer Berlin Heidelberg.

Halldórsson, B. V., and D. F. Gudbjartsson. 2011b. "An algorithm for detecting high frequency copy number polymorphisms using SNP arrays." *J Comput Biol* (School of Science)(Engineering, Reykjavik University, Reykjavik, Iceland. bjarnivh@ru.is) 18, no. 8 (): 955–966.

Halushka, M. K., J. B. Fan, K. Bentley, L. Hsie, N. Shen, A. Weder, R. Cooper, R. Lipshutz, and A. Chakravarti. 1999. "Patterns of single-nucleotide polymorphisms in candidate genes for blood-pressure homeostasis." *Nat Genet* (Department of Genetics)(Center for Human Genetics, Case Western Reserve University School of Medicine)(University Hospitals of Cleveland, Ohio 44106, USA.) 22, no. 3 (): 239–247.

Hart, W. E., C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola. 2017. *Pyomooptimization modeling in python*. Second. Ed. by Springer. Vol. 67. Springer Science & Business Media.

Helman, E., M. S. Lawrence, C. Stewart, C. Sougnez, G. Getz, and M. Meyerson. 2014. "Somatic retrotransposition in human cancer revealed by whole-genome and exome sequencing." *Genome Res* (Harvard-MIT Division of Health Sciences & Technology, Cambridge, Massachusetts 02139, USA; Broad Institute of MIT)(Harvard, Cambridge, Masachusetts 02142, USA;) 24, no. 7 (): 1053–1063.

Helmuth, L. 2001. "Map of the Human Genome 3.0." *Science* 293 (5530): 583b–585b.

Hoehe, M. R., K. Kopke, B. Wendel, K. Rohde, C. Flachmeier, K. K. Kidd, W. H. Berrettini, and G. M. Church. 2000. "Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence." *Hum Mol Genet* (Genome Research, Max-Delbruck-Center

for Molecular Medicine, Robert-Rossle-Strasse 10, D-13092 Berlin, Germany. mhoehe@mdc-berlin.de) 9, no. 19 (): 2895–2908.

Hormozdiari, F., M. K. Konkel, J. Prado-Martinez, G. Chiatante, I. H. Herraez, J. A. Walker, B. Nelson, C. Alkan, P. H. Sudmant, J. Huddleston, C. R. Catacchio, A. Ko, M. Malig, C. Baker, G. A. Genome Project, T. Marques-Bonet, M. Ventura, M. A. Batzer, and E. E. Eichler. 2013. "Rates and patterns of great ape retrotransposition." *Proceedings of the National Academy of Sciences* 110 (33): 13457–13462.

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

Hugot, J. P., M. Chamaillard, H. Zouali, S. Lesage, J. P. Cezard, J. Belaiche, S. Almer, C. Tysk, C. A. O'Morain, M. Gassull, V. Binder, Y. Finkel, A. Cortot, R. Modigliani, P. Laurent-Puig, C. Gower-Rousseau, J. Macry, J. F. Colombel, M. Sahbatou, and G. Thomas. 2001. "Association of NOD2 leucine-rich repeat variants with susceptibility to Crohn's disease." *Nature* (Fondation Jean Dausset CEPH, 27 rue J. Dodu 75010 Paris, France.) 411, no. 6837 (): 599–603.

Kelley, D. R., D. G. Hendrickson, D. Tenen, and J. L. Rinn. 2014. "Transposable elements modulate human RNA abundance and splicing via specific RNA-protein interactions." *Genome Biology* (London) 15 (12): 537.

Korbel, J. O., A. Abyzov, X. J. Mu, N. Carriero, P. Cayting, Z. Zhang, M. Snyder, and M. B. Gerstein. 2009. "PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data." *Genome Biol* (Gene Expression Unit, European Molecular Biology Laboratory (EMBL), Meyerhofstr, Heidelberg, 69117, Germany. korbel@embl.de) 10 (2): R23.

Kremling, A., and J. Saez-Rodriguez. 2007. "Systems biology–an engineering perspective." *J Biotechnol* (Systems Biology Group, Max-Planck-Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany. kremling@mpi-magdeburg.mpg.de) 129, no. 2 (): 329–351.

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

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

Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, and W. FitzHugh. 2001. "Initial sequencing and analysis of the human genome." *Nature* 409.

Larrañaga, P., B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles. 2006. "Machine learning in bioinformatics." *Briefings in Bioinformatics*: 86–112.

Li, W. H., and L. A. Sadler. 1991. "Low nucleotide diversity in man." *Genetics* (Center for Demographic)(Population Genetics, University of Texas Health Science Center, Houston 77225.) 129, no. 2 (): 513–523.

Lu, X., T. Niu, and J. S. Liu. 2003. "Haplotype Information and Linkage Disequilibrium Mapping for Single Nucleotide Polymorphisms." *Genome Research* 13:2112–2117.

Manley, L. J., D. Ma, and S. S. Levine. 2016. "Monitoring Error Rates In Illumina Sequencing." *J Biomol Tech* (BioMicro Center, Department of Biology, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA.) 27, no. 4 (): 125–128.

Marchini, J., D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. S. Qin, H. M. Munro, G. R. Abecasis, and P. Donnelly. 2006. "A comparison of phasing algorithms for trios and unrelated individuals." *American Journal of Human Genetics* 78:437–450.

Mark, A. B., and L. D. Prescott. 2002. "Alu repeats and human genomic diversity." *Nat Rev Genet* (Department of Biological Sciences, Biological Computation)(Visualization Center, Louisiana State University, 202 Life Sciences Building, Baton Rouge, Louisiana 70803, USA. mbatzer@lsu.edu) 3, no. 5 (): 370–379.

Marshall, E. 1999. "Drug Firms to Create Public Database of Genetic Mutations." *Science* 284 (5413): 406–407.

Martin, R. K. 1999. *Large Scale Linear and Integer Optimization: A Unified Approach*. Springer US.

Matsumoto, M., and T. Nishimura. 1998. "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator." *ACM Transactions on Modeling and Computer Simulation* 8 (1): 3–30.

McCarroll, S. A., F. G. Kuruvilla, J. M. Korn, S. Cawley, J. Nemesh, A. Wysoker, M. H. Shapero, P. I. W. de Bakker, J. B. Maller, A. Kirby, A. L. Elliott, M. Parkin, E. Hubbell, T. Webster, R. Mei, J. Veitch, P. J. Collins, R. Handsaker, S. Lincoln, M. Nizzari, J. Blume, K. W. Jones, R. Rava, M. J. Daly, S. B. Gabriel, and D. Altshuler. 2008. "Integrated detection and population-genetic analysis of SNPs and copy number variation." *Nature Genetics* 40, no. 10 (): 1166–1174.

McClellan, J., and M. C. King. 2010. "Genetic Heterogeneity in Human Disease." *Cell* 141:210–217.

Michael, R. G., and S. J. David. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Momma, K., R. Matsuoka, and A. Takao. 1999. "Aortic Arch Anomalies Associated with Chromosome 22q11 Deletion (CATCH 22)," 20 (2): 97–102.

Nemhauser, G. L., and L. A. Wolsey. 1989. "Optimization." In *Handbooks in Operations Research and Management Science*, ed. by G. L. Nemhauser, A. H. G. R. Kan, and M. J. Todd. Elsevier, North-Holland.

Nistico, L., R. Buzzetti, L. E. Pritchard, B. Van der Auwera, C. Giovannini, E. Bosi, M. T. Larrad, M. S. Rios, C. C. Chow, C. S. Cockram, K. Jacobs, C. Mijovic, S. C. Bain, A. H. Barnett, C. L. Vandewalle, F. Schuit, F. K. Gorus, R. Tosi, P. Pozzilli, and J. A. Todd. 1996. "The CTLA-4 gene region of chromosome 2q33 is linked to, and associated with, type 1 diabetes. Belgian Diabetes Registry." *Hum Mol Genet* (Istituto Clinica Medica II, University of Rome La Sapienza, Italy.) 5, no. 7 (): 1075–1080.

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

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

Ogilvie, C. M., J. W. Ahn, K. Mann, R. G. Roberts, and F. Flinter. 2009. "A novel deletion in proximal 22q associated with cardiac septal defects and microcephaly: a case report." *Molecular Cytogenetics* 2:9–9.

Ogura, Y., D. K. Bonen, N. Inohara, D. L. Nicolae, F. F. Chen, R. Ramos, H. Britton, T. Moran, R. Karaliuskas, R. H. Duerr, J. P. Achkar, S. R. Brant, T. M. Bayless, B. S. Kirschner, S. B. Hanauer, G. Nunez, and J. H. Cho. 2001. "A frameshift mutation in NOD2 associated with susceptibility to Crohn's disease." *Nature* (Department of Pathology)(Comprehensive Cancer Center, The University of

Michigan Medical School, Ann Arbor, Michigan 48109, USA.) 411, no. 6837 (): 603–606.

Oosten, M., J. H. G. C. Rutten, and F. C. R. Spieksma. 2001. "The clique partitioning problem: Facets and patching facets." *Networks* 38 (4): 209–226.

Ozaki, K., Y. Ohnishi, A. Iida, A. Sekine, R. Yamada, T. Tsunoda, H. Sato, H. Sato, M. Hori, Y. Nakamura, and T. Tanaka. 2002. "Functional SNPs in the lymphotoxin-alpha gene that are associated with susceptibility to myocardial infarction." *Nat Genet* (Laboratory for Cardiovascular Diseases, SNP Research Center, The Institute of Physical)(Chemical Research (RIKEN), 4-6-1, Shirokanedai, Minato-ku, Tokyo 108-8639, Japan.) 32, no. 4 (): 650–654.

Paolella, G., M. A. Lucero, M. H. Murphy, and F. E. Baralle. 1983. "The Alu family repeat promoter has a tRNA-like bipartite structure." *EMBO J* 2.

Pe'er, I., T. Pupko, R. Shamir, and R. Sharan. 2004. "Incomplete Directed Perfect Phylogeny," *SIAM Journal on Computing* 33, no. 3 (): 590–607.

Pennacchio, M. Olivier, J. A. Hubacek, J. C. Cohen, D. R. Cox, J. C. Fruchart, R. M. Krauss, and E. M. Rubin. 2001. "An apolipoprotein influencing triglycerides in humans and mice revealed by comparative sequencing." *Science* (Genome Sciences Department, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA.) 294, no. 5540 (): 169–173.

Prescott, L. D., and A. B. Mark. 1999. "Alu repeats and human disease." *Mol Genet Metab* (Department of Environmental Health Sciences, Tulane University Medical Center, 1430 Tulane Avenue, New Orleans, Louisiana, 70112, USA. pdeinin@tcs.tulane.edu) 67, no. 3 (): 183–193.

Puvabanditsin, S., M. S. Nagar, M. Joshi, G. Lambert, E. Garrow, and E. Brandsma. 2010. "Microdeletion of 16p11.2 associated with endocardial fibroelastosis." *Am J Med Genet A* (Department of Pediatrics, UMDNJ-RWJ Medical School, New Brunswick, NJ 08903, USA.) 152A, no. 9 (): 2383–2386.

Qian, Y., B. Kehr, and B. V. Halldórsson. 2015. "PopAlu: population-scale detection of Alu polymorphisms." *PeerJ* (Bioinformatics Research Center, Aarhus University , Aarhus , Denmark.) 3:e1269.

Rioux, J. D., M. J. Daly, M. S. Silverberg, K. Lindblad, H. Steinhart, Z. Cohen, T. Delmonte, K. Kocher, K. Miller, S. Guschwan, E. J. Kulbokas, S. O'Leary, E. Winchester, K. Dewar, T. Green, V. Stone, C. Chow, A. Cohen, D. Langelier, G. Lapointe, D. Gaudet, J. Faith, N. Branco, S. B. Bull, R. S. McLeod, A. M. Griffiths, A. Bitton, G. R. Greenberg, E. S. Lander, K. A. Siminovitch, and T. J. Hudson. 2001. "Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to Crohn disease." *Nat Genet* (Whitehead Institute/Massachusetts Institute of Technology, Center for Genome Research, Cambridge, Massachusetts, USA.) 29, no. 2 (): 223–228.

Risch, N., and K. Merikangas. 1996. "The future of genetic studies of complex human diseases." *Science* 273:1516–1517.

Rivadeneira, F., U. Styrkarsdottir, K. Estrada, B. Halldorsson, Y. Hsu, J. B. Richards, M. C. Zillikens, F. Kavvoura, N. Amin, Y. Aulchenko, L. Cupples, P. Deloukas, S. Demissie, E. Grundberg, A. Hofman, A. Kong, D. Karasik, J. van Meurs, B. Oostra, T. Pastinen, H. A. P. Pols, G. Sigurdsson, N. Soranzo, G. Thorleifsson, U. Thorsteindottir, F. Williams, S. Wilson, Y. Zhou, S. Ralston, C. van Duijn, T. Spector, D. Kiel, K. Stefansson, J. Ioannidis, A. G. Uitterlinden, and I. behalf of the GEFOS Consortium. 2009. "Twenty loci associated with bone mineral density identified by large-scale meta-analysis of genome-wide association datasets," *Bone* 44 (): S230–S231.

Salem, A.-H., D. A. Ray, J. Xing, P. A. Callinan, J. S. Myers, D. J. Hedges, R. K. Garber, D. J. Witherspoon, L. B. Jorde, and M. A. Batzer. 2003. "Alu elements and hominid phylogenetics." *Proc Natl Acad Sci U S A* (Department of Biological Sciences, Biological Computation)(Visualization Center, Louisiana State University, 202 Life Sciences Building, Baton Rouge, LA 70803, USA.) 100, no. 22 (): 12787–12791.

Schrijver, A. 2003. *Combinatorial optimization*.

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

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

Shinawi, M., T. Sahoo, B. Maranda, S. A. Skinner, C. Skinner, C. Chinault, R. Zascavage, S. U. Peters, A. Patel, R. E. Stevenson, and A. L. Beaudet. 2011. "11p14.1 microdeletions associated with ADHD, autism, developmental delay, and obesity." *Am J Med Genet A* (Department of Molecular)(Human Genetics, Baylor College of Medicine, Houston, Texas 77030, USA.) 155A, no. 6 (): 1272–1280.

Sontag, E. D. 2005. "Molecular Systems Biology and Control." *European Journal of Control* 11 (4): 396–435.

Sorek, R., G. Ast, and D. Graur. 2002. "Alu-Containing Exons are Alternatively Spliced." *Genome Research* 12, no. 7 (): 1060–1067.

Speed, T., H. Huang, E. Corona, B. Raphael, and E. Eskin. 2007. "Identification of Deletion Polymorphisms from Haplotypes," 4453:354–365. Springer Berlin Heidelberg.

Stefansson, H., D. Rujescu, S. Cichon, O. P. H. Pietilainen, A. Ingason, S. Steinberg, R. Fossdal, E. Sigurdsson, T. Sigmundsson,

J. E. Buizer-Voskamp, T. Hansen, K. D. Jakobsen, P. Muglia, C. Francks, P. M. Matthews, A. Gylfason, B. V. Halldorsson, D. Gudbjartsson, T. E. Thorgeirsson, A. Sigurdsson, A. Jonasdottir, A. Jonasdottir, A. Bjornsson, S. Mattiasdottir, T. Blondal, M. Haraldsson, B. B. Magnusdottir, I. Giegling, H.-J. Moller, A. Hartmann, K. V. Shianna, D. Ge, A. C. Need, C. Crombie, G. Fraser, N. Walker, J. Lonnqvist, J. Suvisaari, A. Tuulio-Henriksson, T. Paunio, T. Toulopoulou, E. Bramon, M. Di Forti, R. Murray, M. Ruggeri, E. Vassos, S. Tosato, M. Walshe, T. Li, C. Vasilescu, T. W. Muhleisen, A. G. Wang, H. Ullum, S. Djurovic, I. Melle, J. Olesen, L. A. Kiemeney, B. Franke, C. Sabatti, N. B. Freimer, J. R. Gulcher, U. Thorsteinsdottir, A. Kong, O. A. Andreassen, R. A. Ophoff, A. Georgi, M. Rietschel, T. Werge, H. Petursson, D. B. Goldstein, M. M. Nothen, L. Peltonen, D. A. Collier, D. St Clair, and K. Stefansson. 2008. "Large recurrent microdeletions associated with schizophrenia." *Nature* 455, no. 7210 (): 232–236.

Stefansson, E. Sigurdsson, V. Steinthorsdottir, S. Bjornsdottir, T. Sigmundsson, S. Ghosh, J. Brynjolfsson, S. Gunnarsdottir, O. Ivarsson, T. T. Chou, O. Hjaltason, B. Birgisdottir, H. Jonsson, V. G. Gudnadottir, E. Gudmundsdottir, A. Bjornsson, B. Ingvarsson, A. Ingason, S. Sigfusson, H. Hardardottir, R. P. Harvey, D. Lai, M. Zhou, D. Brunner, V. Mutel, A. Gonzalo, G. Lemke, J. Sainz, G. Johannesson, T. Andresson, D. Gudbjartsson, A. Manolescu, M. L. Frigge, M. E. Gurney, A. Kong, J. R. Gulcher, H. Petursson, and K. Stefansson. 2002. "Neuregulin 1 and susceptibility to schizophrenia." *Am J Hum Genet* (deCODE Genetics, Reykjavik, Iceland. kstefans@decode.is.) 71, no. 4 (): 877–892.

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

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

Stranger, B. E., E. A. Stahl, and T. Raj. 2011. "Progress and promise of genome-wide association studies for human complex trait genetics." *Genetics* (Division of Genetics, Brigham)(Women's Hospital, Boston, Massachusetts 02115, USA. bstranger@rics.bwh.harvard.edu) 187, no. 2 (): 367–383.

Strittmatter, W. J., and A. D. Roses. 1996. "Apolipoprotein E and Alzheimer's disease." *Annual Reviews - Neuroscience* 19:53–77.

Styrkarsdottir, U., B. V. Halldórsson, S. Gretarsdottir, D. F. Gudbjartsson, G. B. Walters, T. Ingvarsson, T. Jonsdottir, J. Saemundsdottir, J. R. Center, T. V. Nguyen, Y. Bagger, J. R. Gulcher, J. A. Eisman, C. Christiansen, G. Sigurdsson, A. Kong, U. Thorsteinsdottir, and K. Stefansson. 2008. "Multiple genetic loci for bone mineral density and fractures." *N Engl J Med* (deCODE Genetics, Reykjavik, Iceland.) 358, no. 22 (): 2355–2365.

Sutherland, W. J. 2005. "The best solution." *Nature* (Centre for Ecology, Evolution)(Conservation, School of Biological Sciences, the University of East Anglia, Norwich NR4 7TJ, UK.) 435, no. 7042 (): 569.

Sveinbjörnsson, J. I., and B. V. Halldórsson. 2012. "PAIR: polymorphic Alu insertion recognition." *BMC Bioinformatics* 13, no. Suppl 6 (): S7.

Terwilliger, J. D., and K. M. Weiss. 1998. "Linkage disequilibrium mapping of complex disease: fantasy or reality?" *Curr Opin Biotechnol* 9, no. 6 (): 578–594.

Ullu, E., and C. Tschudi. 1984. "Alu sequences are processed 7SL RNA genes."

Van Eerdewegh, P., R. D. Little, J. Dupuis, R. G. Del Mastro, K. Falls, J. Simon, D. Torrey, S. Pandit, J. McKenny, K. Braunschweiger, A. Walsh, Z. Liu, B. Hayward, C. Folz, S. P. Manning, A. Bawa, L. Saracino, M. Thackston, Y. Benchekroun, N. Capparell, M. Wang, R. Adair, Y. Feng, J. Dubois, M. G. FitzGerald, H. Huang, R. Gibson, K. M. Allen, A. Pedan, M. R. Danzig, S. P. Umland, R. W. Egan, F. M. Cuss, S. Rorke, J. B. Clough, J. W. Holloway, S. T. Holgate, and T. P. Keith. 2002. "Association of the ADAM33 gene with asthma and bronchial hyperresponsiveness." *Nature* (Genome Therapeutics Corporation, 100 Beaver St, Waltham, Massachusetts 02453, USA.) 418, no. 6896 (): 426–430.

Venter, J. C., et al. 2001. "The Sequence of the Human Genome." *Science* 291 (5507): 1304–1351.

Victor, A. A. 2005. *Parsimony, Phylogeny, and Genomics*. Ed. by A. A. Victor. Oxford University Press.

Wang, D. G., J. B. Fan, C. J. Siao, A. Berno, P. Young, R. Sapolsky, G. Ghandour, N. Perkins, E. Winchester, J. Spencer, L. Kruglyak, L. Stein, L. Hsie, T. Topaloglou, E. Hubbell, E. Robinson, M. Mittmann, M. S. Morris, N. Shen, D. Kilburn, J. Rioux, C. Nusbaum, S. Rozen, T. J. Hudson, R. Lipshutz, M. Chee, and E. S. Lander. 1998. "Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome." *Science* (Whitehead Institute for Biomedical Research, Nine Cambridge Center, Cambridge, MA 02142, USA.) 280, no. 5366 (): 1077–1082.

Wang, J., L. Song, M. K. Gonder, S. Azrak, D. A. Ray, M. A. Batzer, S. A. Tishkoff, and P. Liang. 2006. "Whole genome computational comparative genomics: A fruitful approach for ascertaining Alu insertion polymorphisms." *Gene* (Department of Cancer Genetics, Roswell Park Cancer Institute, Elm)(Carlton Streets, Buffalo, NY 14263, USA.) 365 (): 11–20.

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

Wolkenhauer, O., M. Ullah, P. Wellstead, and K.-H. Cho. 2005. "The dynamic systems approach to control and regulation of intracellular networks." *FEBS Lett* (Department of Computer Science, Systems Biology)(Bioinformatics Group, University of Rostock, Albert Einstein Str. 21, 18059 Rostock, Germany. olaf.wolkenhauer@uni-rostock.de) 579, no. 8 (): 1846–1853.

Wolsey, L. A. 1998. *Integer Programming*. Wiley-Interscience, NY.

Zhang, W., A. Edwards, W. Fan, P. Deininger, and K. Zhang. 2011. "Alu distribution and mutation types of cancer genes." *BMC Genomics* 12 (1): 157.

Zhang, X. S., R. S. Wang, L. Y. Wu, and L. Chen. 2006. "Models and algorithms for haplotyping problem." *Current Bioinformatics* 1:105–114.