UCLouvain

LOUVAIN
School of Management

Louvain School of Management

Reconstructing phylogenies from genotype sequence collections: Merging the Pure Parsimony Haplotyping problem with the Haplotype Phylogeny problem

Author : Claire Verstegen
Supervisor(s) : Daniele Catanzaro
Academic year 2019-2020

Reconstructing phylogenies from genotype sequence collections: Merging the Pure Parsimony Haplotyping problem with the Haplotype Phylogeny problem

Claire Verstegen

June 2, 2020

Abstract

The information provided by haplotype and phylogenetic estimation methods is of central importance e.g., in the diagnosis of the genetic causes associated to major human diseases as well as in the design of new therapeutical targets and personalized treatments. Hence, the scientific community has devoted in recent times increasing research efforts on the development of more and more capable predictive models for haplotyping and phylogeny estimation. So far, the usual approach to the modeling of haplotyping and phylogeny estimation consisted in considering both problems separately and one (haplotyping) after the other (phylogeny estimation). This thesis investigates how to improve this approach by considering both problems simultaneously. In particular, in a first attempt we will focus on the merging of both problems with respect to the parsimony criterion of haplotype and phylogeny estimation and we will propose some alternative integer linear programming models to solve this new merged problem. We compare the performances of these formulations on a set of small real instances of both problems and we analyze the corresponding results. Our analysis shows that the proposed merging constitutes a viable way to carry out the analysis of fine-scale genetic data and definitely warrants additional research efforts.

Keywords: integer programming, maximum parsimony, haplotype inference, phylogeny estimation

1 Introduction

In 1990, an international scientific research project called the *Human Genome Project* (HGP) (National Human Genome Institute, 2019; Collins and Galas, 1993) was launched, with very ambitious goals: among others, the project aimed at making the complete *human genome* sequence freely available for researchers, improving the quality and efficiency of DNA sequencing technology while reducing its cost, studying the natural sequence variation among the human genome, developing technology for functional genomics in order to support health care researches, and supporting bioinformatics and computational biology (Collins et al., 1998). This major collaborative project was completed in 2003, making the world able to unravel the complete genetic blueprint that is responsible for building human beings (National Human Genome Institute, 2019). The completion of this project had huge consequences on different research fields, including structural biology, health care, bioinformatics, and computational biology (Burley et al., 1999; Collins and McKusick, 2001).

Among other findings, the HGP helped people understand the role of heredity in most human illnesses, making their diagnosis and treatment easier thanks to genomic medicine (Collins and McKusick, 2001). In parallel, another important result of this project enabled the demonstration of great similarities between the two copies constituting any human DNA. In fact, it has been shown that less than 1% of the nucleotides forming the genomes of any two people are different. Consequently, these differences (or *polymorphisms*) contribute to the variation of individuals' phenotype, and are therefore risk factors that could explain some genetic diseases (Catanzaro et al., 2010; The International SNP Map Working Group, 2001).

A variant site of the genome is called a *Single Nucleotide Polymorphism* (SNP). For the majority of the population, we observe a SNP to be a certain *allele* (a possible value for a gene); and we observe a different allele for a minority of it. A haplotype is therefore constituted by the values taken by the set of SNPs on a region of a chromosome (Catanzaro et al., 2010). To detect if an individual is affected by a genetic disease, it is useful to have a look at his or her *haplotypes* (The International SNP Map Working Group, 2001; Johnson et al., 2001). The diploid nature of human organisms implies that their genome is comprised of pairs of

haplotypes: one inherited from their father and one from their mother. We call *genotype* the combination of a pair of haplotypes (Catanzaro and Labbé, 2009).

The conclusions of the HGP, showing the important potential of this subject, led to a great deal of research about haplotypes estimation all around the globe (Fan et al., 2011; Gusfield, 2003; Niu et al., 2002). Indeed, different methods can be used to find the haplotypes of an individual. Basically, the two main approaches are the experimental and the computational ones. The experimental approach is not applicable in any case: it requires the availability of family-based genetic information, and implies advanced molecular isolation strategies. This approach is therefore very expensive and is in some cases an inefficient use of the resources (Stephens et al., 2001), making researchers look towards computational methods. As for them, these methods imply to solve an optimization problem, using a set of genotypes as input, and requiring that the optimal set of haplotypes found can generate the input of genotypes (Catanzaro and Labbé, 2009; Gusfield, 2001).

Haplotyping can be applied to many fields, and is not limited to the detection of genetic diseases. We can cite some other interesting uses of haplotypes. A first example is *Human Leukocyte Antigen* (HLA) matching, which allows for an assessment of the compatibility between transplant recipients and donors. Haplotypes are also helpful in the field of pharmacogenomics: by predicting the drug response of individuals, it can lead to better treatment of certain diseases (Crawford and Nickerson, 2005). We can also use them in functional genomics, when a gene is the cause of the response of an organism to therapies (Catanzaro et al., 2010). On another note, they can be used to infer the history of populations through the construction of genealogic trees from their haplotype sequences (Wilson et al., 2003; Long et al., 1990).

It is worth noting that the progress in genome research, creating more and better quality data for studies and research (Collins et al., 1998), led to improvements in the field of molecular phylogenetics. Molecular phylogenetics aims to studying how organisms using molecular sequence or structured data (DNA, RNA, or proteins) evolve over time, how they develop and what the relationships among them are. Any phylogenetic analysis is based on the assumption that the organisms studied are related, evolutionarily speaking. This type of study depicts relationships through a tree (Atri and Lichtarge, 2018; Bleidorn, 2017). These trees are called phylogenies and are mathematical models of evolution (Warnow, 2013).

To construct a phylogeny from a set of sequences, there exist different methods (Yang and Rannala, 2012; Warnow, 2019). However, all of them are characterized by three elements: a criterion to select which phylogeny is better than the others, a formula to compute the length of branches, and an algorithm to explore the possibilities (Paradis, 2012). One method, better suited to combinatorial approaches, implies to solve a *Steiner tree problem*. This class of problems comprises combinatorial optimization problems and aims at computing a weighted tree that contains all *terminals* (or *nodes*) given as input, while minimizing the total size of the tree, i.e. the sum of distances between connected terminals (Schwartz, 2019). Under certain conditions, the application of this problem on the solving of sequences of binary taxa such as haplotypes comes down to solving a *Steiner tree problem in the hypercube* (Blelloch et al., 2006).

These scientific advances led many researchers to address the following problem: estimate phylogenies from aligned haplotype sequences. Once more, the applications are multiple. Indeed, the knowledge of accurate evolutionary relationships that exist among organisms allows to interpret meaning from data studied. Like haplotyping, molecular phylogenetics influenced many fields: medical research, epidemiology, drug discovery, population dynamics, etc. More concretely, it allowed for the study of particular viruses, leading to practical findings regarding for instance influenza A or HIV virus, identifying some viruses as *Severe Acute Respiratory Syndrome* (SARS), etc. Other studies about genes and proteins led to a better understanding of the way they affect each other and their response to each other's presence (Catanzaro et al., 2013b; Boldt et al., 2010).

The logical continuation of these researches leads to the following idea: estimating from a set of genotypes not only the haplotypes, but also the evolutionary relationships that relate these haplotypes. A way to do so is first solving an optimization problem in order to estimate a set of haplotypes from a set of genotypes; and then solving a Steiner tree problem in the hypercube to estimate a phylogeny from them. To go a step further, we can merge these two problems. We should estimate a phylogeny of haplotypes from a set of genotypes through an optimization problem. This problem would require to simultaneously find the minimum number of haplotypes that explain the given set of genotypes and minimizing the length of the tree connecting all haplotypes.

2 Decision criterion

There is no general way to empirically validate a set of haplotypes over the other candidates, nor is there a way to determine the true phylogeny for these haplotypes. This involves choosing a criterion allowing to select a candidate when solving the optimization problem. The literature proposes different selection criteria, each of them based on different hypotheses but characterized by biological motivations (Catanzaro et al., 2010; Catanzaro, 2010). The chosen criterion impacts directly the objective function of the model, creating different families of problems. In the literature, the criteria that are the most widely used for haplotyping as for phylogenies estimation are the *Maximum Likelihood* and the *Maximum Parsimony* (Catanzaro and Labbé, 2009; Catanzaro, 2010).

The criterion that we have chosen in order to develop our optimization model is the Maximum Parsimony criterion, sometimes called *Ockham's razor* (Steel and Penny, 2000). The idea behind this criterion is that the explanation of a phenomenon that requires the fewest assumptions should be preferred (Catanzaro et al., 2010). In our model, we can translate this idea into the following principles: the optimal set of haplotypes for a given set of genotypes is the one with the smallest cardinality; and the optimal phylogeny is the one with the smallest number of evolutionary changes (Catanzaro and Labbé, 2009; Paradis, 2012).

This choice seems relevant for several reasons. Apart from its simplicity, and therefore its ease of understanding and implementation (Bleidorn, 2017; Steel and Penny, 2000), the most important one is that under certain conditions, parsimony is a good approximation of the way taxa evolve. According to some researchers, evolution proceeds by *small* changes. Approximating them with *smallest* changes is valid for molecular regions that are well-conserved, where mutations do not occur often and if so, not repeatedly on the same variant site. Parsimony criterion is therefore reasonable for intraspecies phylogenetics, as the expected number of mutations is close to zero (Catanzaro et al., 2013b). Moreover, the researches on haplotypes have shown that the number of distinct haplotypes actually observed in a population is smaller than the total number of possibilities (Catanzaro et al., 2010). Finally, and as opposed to the Maximum Likelihood criterion, the Maximum Parsimony is the best choice for applications where it is impractical or impossible to develop stochastic models with many parameters (Warnow, 2013).

Despite these favorable arguments, it is necessary to mention that Maximum Parsimony is *statistically inconsistent* in some cases. Statistical consistency is a desired property, stating that as the amount of data increases, the solution considered as optimal according to the criterion approaches the real solution (Catanzaro, 2010). In other words, a model is consistent if its probability to find the true solution converges to 1 as the sequence tends to the infinite (Steel and Penny, 2000). For this reason, the advocates of Maximum Likelihood method have widely criticized the Maximum Parsimony criterion. However, we need to bring two nuances to this debate. First, statistical consistency is hard to prove for broad models: some can be consistent for a class of problems, and inconsistent for others. This propriety is thus often demonstrated for models under specific conditions, and Maximum Parsimony is statistically consistent in various cases (Steel and Penny, 2000; Rzhetsky and Nei, 1992; DeBry, 1992). Then, we recall that the definition of statistical consistency is based on the idea of computation on samples with a very large number of sequences, which does not correspond to the real length of sequences studied in phylogenetics. On shorter samples, there is no evidence that statistically consistent methods outperform inconsistent ones (Steel and Penny, 2000).

In light of this information, we can formulate the problem that this master's thesis intends to solve. In this paper, we present an *Integer Linear Program* (ILP) model for solving the following problem: finding the most parsimonious phylogeny from the set of haplotypes necessary to resolve an input of genotypes.

This problem is therefore a variant of the haplotype estimation problem under Maximum Parsimony, to which we add a tree constraint. It is based on two problems widely studied in the literature, the *Pure Parsimony Haplotyping* (PPH) problem (Catanzaro and Labbé, 2009) and the *Most Parsimonious Phylogeny Estimation Problem* (MPPEP) (Catanzaro et al., 2013b). It is worth mentioning that as both problems are $\mathcal{NP}$-Hard (Catanzaro and Labbé, 2009; Warnow, 2013), the problem that we will treat is $\mathcal{NP}$-Hard too.

We present a formulation for this problem inspired by the formulations found in Catanzaro and Labbé (2009) and Catanzaro et al. (2013b). The core idea of this model is to minimize the size of a phylogeny made of haplotypes estimated from genotype data.

Instance of PPH

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

A solution (among others)

Haplotypes	SNP			
Haplotype 1	0	0	1	1
Haplotype 2	1	1	1	0
Haplotype 3	1	0	0	1
Haplotype 4	1	0	0	0
Haplotype 5	0	0	0	0

Conflation				
Genotype 1	=	Haplotype 2	$\oplus$	Haplotype 4
Genotype 2	=	Haplotype 1	$\oplus$	Haplotype 5
Genotype 3	=	Haplotype 1	$\oplus$	Haplotype 3

Figure 1: Graphical representation of an instance of PPH and a possible solution.

3 Notation

Before diving into the modeling of the problem, we introduce some notation that will prove useful throughout this work. Remember that the focus here is not on the whole genome but only on the SNPs: the nucleotide sites of a chromosome region for which there is variability among a population. In haplotypes, SNPs can take two different values: if the observed allele is a minority within the population, it is called the *minor allele* or *mutant type* and is encoded as '1'; while if the observed allele is a majority, it is called the *major allele* or *wild type* and is encoded as '0'. As a result, a set of m alleles forms a haplotype, i.e. a string of length m over an alphabet $\Sigma = \{0, 1\}$.

Humans are diploid organisms: each gene is made up of two alleles (one inherited from the father and one from the mother)[1], and can therefore be either homozygous: both alleles are of type '0' or of type '1'; or heterozygous: an allele is of type '0' and the other is of type '1'. A set of m genes of an individual forms his or her genotype, i.e. a string of length m over an alphabet $\Sigma = \{0, 1, 2\}$. In this notation, the symbols '0' and '1' denote homozygous sites and the symbol '2' denotes a heterozygous site.

Another important notation to define is the operator sum $\oplus$: the operator allowing to combine two haplotypes to obtain a genotype. Given a set K of n genotypes, denote $P = \{1, 2, \ldots, m\}$ as the set of alleles, and $g_k(p), p \in P$, as the pth allele of genotype $g_k, k \in K$. According to this operator, the sum $\oplus$ among the haplotypes h_i and h_j is the genotype g_k, whose pth entry is $h_i(p)$ if $h_i(p) = h_j(p)$ and 2 otherwise. For example, the sum $\oplus$ of the haplotypes $< 1, 0, 1, 0 >$ and $< 1, 1, 0, 0 >$ is the genotype $< 1, 2, 2, 0 >$. We say that a pair of haplotypes h_i and h_j resolves a genotype g_k if $h_i \oplus h_j = g_k$.

Haplotyping a set of genotypes under Maximum Parsimony implies to solve the *Pure Parsimony Haplotyping* (PPH) problem (Catanzaro and Labbé, 2009):

Problem. *PPH*
Given a set K of n genotypes, having m SNPs each, find the minimum set I of haplotypes such that for each genotype $g_k, k \in K$ there exists a pair of haplotypes $\{h_i, h_j\} \in I$ resolving g_k.

For example, you can find an instance of PPH and its corresponding solution in Figure 1. However, our problem is not limited to haplotyping as we have added a tree constraint to it. The concept of tree is strongly linked to that of *adjacency*: we say that two haplotypes are adjacent if the distance between them, i.e. $d_{h_i,h_j} = \sum_{p \in P} |h_i(p) - h_j(p)|$, is equal to 1. Biologically, we consider that if two distinct haplotypes are adjacent, we can deduce that one of them evolved from the other by means of mutation of a variant site (Catanzaro et al., 2013b).

From that, we can construct a graph $G = (I, E)$ having as *vertices* (or *nodes*) the set of haplotypes I that solves the set of genotypes K; and as *edges* the links between adjacent haplotypes, the set E. Such

[1] This is not true all the time: *Loss Of Heterozygosity* (LOH) events can occur due to deletion polymorphism. We refer the reader interested in this kind of phenomenon to the article of Catanzaro et al. (2013a).

a graph allows to build a phylogeny T of I, in other words an acyclic subgraph of G. It is here that the definition of adjacency takes all its importance: if two haplotypes h_i and h_j can be considered adjacent only if $d_{h_i,h_j} = 1$, such a phylogeny could not exist if the graph $G = (I, E)$ is not connected, requiring to add haplotypes to the original set (Catanzaro et al., 2013b). In our case, the problem is slightly different, as the set of haplotypes is not a data but a variable to estimate. The necessity for G to be connected has therefore a direct impact on the upper bound of the set of haplotypes I. In the worst case scenario, this set contains all possible haplotypes, which means 2^m haplotypes (with m being the number of SNPs). As this bound grows exponentially, requiring $d_{h_i,h_j} = 1$ to allow adjacency between h_i and h_j slows down the solving of the model for genotypes with a large number of SNPs. For this reason, we decide to relax the constraint in this model, allowing adjacency for haplotypes whose distance separating them exceeds 1. This has as consequence that the edges have to be weighted by the distance they cover. Thanks to this relaxation, the bound of I is limited to $2n$, which is the maximum number of haplotypes necessary to resolve n genotypes.

Then, the problem of finding a phylogeny of I with the haplotypes in I resolving the genotypes in K, satisfying the Maximum Parsimony criterion consists of solving the following optimization problem:

Problem. *Haplotypes estimation with tree constraint under Maximum Parsimony*
Let GEN be a $n \times m$ matrix of genotypes whose generic entry $g_i \in \{0, 1, 2\}$. Find the shortest tree T of a matrix I having the same number of columns as GEN and the minimum number of rows such that for each row in GEN, there exist two rows in I, say h_i and h_j, such that $g_k = h_i \oplus h_j$.

In the next section, we will develop an ILP model able to solve exactly this problem.

4 Model formulation

Denote $u_i, i \in I$, as a decision variable equal to 1 if the ith haplotype in I is considered in the optimal solution, 0 otherwise; $z_i(p), i \in I, p \in P$, as a decision variable having the value of the pth allele of the ith haplotype in I; $y^k_{ij}, k \in K, i, j \in I : i < j$, as a decision variable equal to 1 if the genotype g_k is solved by the pair of haplotypes i, j, 0 otherwise; $x_{ij}, i, j \in I : i < j$ as a decision variable equal to 1 if the ith and jth haplotypes are adjacent in the optimal solution, 0 otherwise; the decision variable $t_{ij}(p) = |z_i(p) - z_j(p)|, p \in P, i, j \in I : i < j$, in other words, $t_{ij}(p)$ is equal to 1 if $z_i(p)$ and $z_j(p)$ are different and 0 otherwise; and finally the decision variable w_{ij} as the weight of the edge between the adjacent haplotypes i, j, with $w_{ij} \in \{0, \dots, m\}, i, j \in I : i < j$.

4.1 Formulation 1

$$\min \sum_{i \in I} u_i + \sum_{i \in I} \sum_{j \in I : j > i} w_{ij}, \tag{1}$$

$$s.t. \;\; z_i(p) \leq u_i, \qquad \forall\, i \in I,\; p \in P, \tag{2}$$

$$y^k_{ij} \leq u_i, \qquad \forall\, k \in K,\; i, j \in I : i < j, \tag{3}$$

$$y^k_{ij} \leq u_j, \qquad \forall\, k \in K,\; i, j \in I : i < j, \tag{4}$$

$$\sum_{i \in I} \sum_{j \in I : j > i} y^k_{ij} = 1, \qquad \forall\, k \in K, \tag{5}$$

$$z_i(p) \geq \sum_{j \in I : j > i} y^k_{ij} + \sum_{j \in I : j < i} y^k_{ji}, \qquad \forall\, i \in I,\; k \in K,\; p \in P : g_k(p) = 1, \tag{6}$$

$$z_i(p) \leq 1 - \left(\sum_{j \in I : j > i} y^k_{ij} + \sum_{j \in I : j < i} y^k_{ji} \right), \qquad \forall\, i \in I,\; k \in K,\; p \in P : g_k(p) = 0, \tag{7}$$

$$z_i(p) + z_j(p) \leq 2 - y^k_{ij}, \qquad \forall\, i, j \in I : i \neq j,\; k \in K,\; p \in P : g_k(p) = 2, \tag{8}$$

$$z_i(p) + z_j(p) \geq y^k_{ij}, \qquad \forall\, i, j \in I : i \neq j,\; k \in K,\; p \in P : g_k(p) = 2, \tag{9}$$

$$t_{ij}(p) \geq z_i(p) - z_j(p), \qquad \forall\, i, j \in I : i \neq j,\; p \in P, \tag{10}$$

$$t_{ij}(p) \geq -z_i(p) + z_j(p), \qquad \forall\, i, j \in I : i \neq j,\; p \in P, \tag{11}$$

$$w_{ij} \leq m \times x_{ij}, \qquad \forall\, i,j \in I : i \neq j, \tag{12}$$

$$w_{ij} \leq \sum_{p \in P} t_{ij}(p), \qquad \forall\, i,j \in I : i \neq j, \tag{13}$$

$$w_{ij} \geq \sum_{p \in P} t_{ij}(p) - (1 - x_{ij}) \times m, \qquad \forall\, i,j \in I : i \neq j, \tag{14}$$

$$x_{ij} \leq u_i, \qquad \forall\, i,j \in I : i \neq j, \tag{15}$$

$$x_{ij} \leq u_j, \qquad \forall\, i,j \in I : i \neq j, \tag{16}$$

$$\sum_{j \in I : j > i} x_{ij} + \sum_{j \in I : j < i} x_{ji} \geq u_i, \qquad \forall\, i \in I, \tag{17}$$

$$\sum_{i \in I} \sum_{j \in I : j > i} x_{ij} = \sum_{i \in I} u_i - 1, \tag{18}$$

$$\sum_{i \in S} \sum_{j \in S : j > i} x_{ij} \leq \sum_{i \in S} u_i - 1, \qquad \forall\, S \subset I : 2 \leq |S| \leq n - 2, \tag{19}$$

$$z_i(p), u_i, y_{ij}^k, x_{ij}, t_{ij}(p) \in \{0,1\}, \qquad \forall\, i,j \in I : i \neq j,\ k \in K,\ p \in P, \tag{20}$$

$$w_{ij} \in \{0, ..., m\}, \qquad \forall\, i,j \in I : i \neq j. \tag{21}$$

The objective function (1) aims at minimizing the size of the phylogeny, i.e. the total number of nodes (or cardinality of I) plus the sum of edges' weights. Constraints (2) impose that the pth allele of haplotype $i \in I$ can assume value 1 only if the haplotype i is considered in the optimal solution to the problem. Constraints (3)-(4) impose that a haplotype can solve a genotype only if it is considered in the optimal solution. Constraints (5) force each genotype $k \in K$ to be solved by exactly one pair of haplotypes $i, j \in I : i \neq j$. It is worth noting that this makes the solving of genotypes without any heterozygous site impossible, as they are resolved by a pair of identical haplotypes. Genotypes constituted by only 0 and 1 should therefore be excluded from the set to resolve. Constraints (6)-(9) represent the sum operation $\oplus$ among haplotypes. Constraints (6) force the pth SNP of the haplotypes that resolve the genotype $k \in K$ to be equal to 1 when $g_k(p) = 1$. In the same way, constraints (7) force the pth SNP of the haplotypes that resolve the genotype $k \in K$ to be equal to 0 when $g_k(p) = 0$. Constraints (8)-(9) impose that exactly one among the pth SNPs of haplotypes $i, j \in I : i \neq j$ can be set to 1 when they resolve together the genotype $g_k, k \in K$ and $g_k(p) = 2$. Constraints (10)-(11) enforce $t_{ij}(p)$ to be equal to 1 if $z_i(p) \neq z_j(p)$; in other words they translate the non-linear constraints $t_{ij}(p) = |z_i(p) - z_j(p)|,\ p \in P,\ i,j \in I : i < j$. In the same idea, constraints (12)-(14) translate linearly the constraints $w_{ij} = x_{ij} \times \sum_{p \in P} t_{ij}(p)$, meaning that the weight of a link is equal to the distance between the haplotypes if they are adjacent in the optimal solution, otherwise this weight is 0. Constraints (15)-(16) impose that edges can exist only between haplotypes that are considered in the optimal solution. Constraints (17) impose that if the haplotype $i \in I$ is considered in the optimal solution, it should be linked to at least one other haplotype. Constraints (18)-(19) are the *Subtour Elimination Constraints* (SEC) (Catanzaro et al., 2013b). Finally, constraints (20)-(21) are integrity constraints.

Constraints (19) are exponentially many so including them in the formulation is impractical. Hence, we implement a heuristic to break subtours. This heuristic can be described as follows. After a first optimization of the objective function, the procedure checks if some haplotypes are isolated from the main tree (i.e. the tree to which the closest haplotype to $< 0, 0, ..., 0 >$ belongs) thanks to a parameter indicating belonging to that tree, starting with one haplotype and spreading between related haplotypes. If some are isolated, it means that there are subtours. Indeed, constraint (18) involves that the number of links equals the number of haplotypes minus one. In a tour, there are as many edges as nodes, which implies that not all nodes can be connected if there is at least one subtour. Consequently, if the procedure finds that some haplotypes are isolated from the others, it breaks the subtour by adding a constraint to the model that imposes the existence of an edge between a haplotype belonging to the main phylogeny and an isolated haplotype. This additional constraint only prevents this single subtour to exist, therefore it may be necessary to add other constraints. Then, we compute a new optimum for the objective function, and the procedure is iterated until the optimal solution does not contain any subtour, which means that all haplotypes are connected in a single tree.

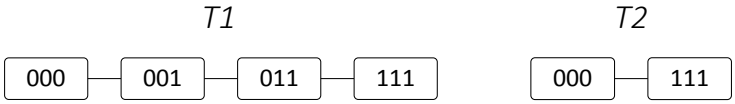

Figure 2: Potential phylogenies from haplotypes resolving $< 2, 2, 2 >$.

4.2 Formulation 2

An alternative formulation for the problem can be obtained by replacing variable y_{ij}^k with a variable y_i^k such that if a haplotype $i \in I$ resolves a genotype $k \in K$, y_i^k is equal to 1, else it is equal to 0. Thanks to that, the number of variables y decreases from $n^2 \times (2n-1)$ to $2n^2$. The new variables imply to change the constraints from the original formulation in which they play a role, i.e. constraints from (3) to (9). They are replaced by:

$$y_i^k \leq u_i, \qquad \forall\ k \in K,\ i \in I, \tag{22}$$

$$\sum_{i \in I} y_i^k = 2, \qquad \forall\ k \in K, \tag{23}$$

$$z_i(p) \geq y_i^k, \qquad \forall\ i \in I,\ k \in K,\ p \in P : g_k(p) = 1, \tag{24}$$

$$z_i(p) \leq 1 - y_i^k, \qquad \forall\ i \in I,\ k \in K,\ p \in P : g_k(p) = 0, \tag{25}$$

$$z_i(p) + z_j(p) \leq 3 - (y_i^k + y_j^k), \qquad \forall\ i, j \in I : i \neq j,\ k \in K,\ p \in P : g_k(p) = 2, \tag{26}$$

$$z_i(p) + z_j(p) \geq y_i^k + y_j^k - 1. \qquad \forall\ i, j \in I : i \neq j,\ k \in K,\ p \in P : g_k(p) = 2. \tag{27}$$

Constraints (22) replace (3)-(4) and impose that a haplotype can resolve a genotype only if it is considered in the optimal solution. Constraints (23) replace (5) and force each genotype to be resolved by exactly 2 different haplotypes. Again, this excludes from the model genotypes that do not have any heterozygous site. Finally, constraints (24)-(27) replace (6)-(9) to translate the sum operator $\oplus$.

4.3 Formulation 3

It is worth reminding that the definition of *adjacency* that we apply in this model is a relaxation with respect to the original one, in order to make the problem solvable in a reasonable time. In the initial understanding of the problem, adjacency is allowed only between haplotypes that differ from each other by only one SNP. However, a simple example shows that the model allowing relaxed adjacency has a tendency to favor solutions with a small number of haplotypes and large weights.

We could take as an example a very simple situation, with a single genotype to resolve: $< 2, 2, 2 >$. We can easily find two haplotypes able to solve it: $< 0, 0, 0 >$ and $< 1, 1, 1 >$. Potential phylogenies from them are displayed in Figure 2. In the first phylogeny, all the edges have a weight equal to 1. In the second one, the unique edge has a weight of 3. The value of the objective function of $T1$, which is composed of 4 haplotypes and 3 edges of weight 1, is equal to 7; while the objective function of $T2$, which is composed of 2 haplotypes and an edge of weight 3, is equal to 5. The models developed in Formulations 1 and 2 would prefer $T2$ over $T1$, which is not desirable in the context of the problem initally stated.

Consequently, to move towards a model that provides as optimal solution a phylogeny that is closer to the one desired, different solutions are to be considered. The first thing that comes to mind is to modify the objective function:

$$\min \sum_{i \in I} \sum_{j \in I : j > i} w_{ij}. \tag{28}$$

When taking only the edges' weights into account, in principle, the model is equally likely to converge on both phylogenies in the example. However, this suffers from a major drawback: removing the number of haplotypes from the function to minimize results in the fact that the same haplotype can be considered several times in the optimal solution. The model ends up with zero weight links, for which it has not been configured. This can lead to errors in the solving. Moreover, this objective function does not guarantee that

the optimal phylogeny obtained will present only edges having a weight of 1. Thus, to determine if this relaxation is really beneficial, it is necessary to develop a different model, in a way that the results and the performance are comparable.

Then, we propose a last formulation to solve the problem stated. In it, we consider that adjacency between two haplotypes $h_i, h_j, i, j \in I$ is possible only if $d_{h_i,h_j} = 1$, in other words if they are similar with exception of one SNP. To take account of this stricter definition, we must modify the model developed in Formulation 1 on which we base ourselves.

First of all, it is important to precise that in this new formulation, the definition of the set I is different. As explained above, the necessity for the graph $G = (I, E)$ to be connected requires the set of haplotypes to be larger. For this reason, the upper bound of I must be defined at 2^m, which represents the total number of different haplotypes that exist having m SNPs, as each site can take two possible values.

In this alternative formulation, all the variables remain the same except one: w_{ij}. Instead of representing the weight of the edge that links haplotypes h_i and h_j, it becomes here the distance between both haplotypes, the number of SNPs that differ from one to another. We can compute it as: $w_{ij} = \sum_{p \in P} t_{ij}(p) = \sum_{p \in P} |z_i(p) - z_j(p)| \forall\ i, j \in I : i \neq j$. This value is therefore independent from the fact that two haplotypes are considered adjacent in the optimal solution. In contrast, it will influence the edges that the model can consider in the phylogeny.

The objective function becomes the following:

$$\min \sum_{i \in I} u_i + \sum_{i \in I} \sum_{j \in I : j > i} x_{ij} \tag{29}$$

Besides, some constraints must be replaced. In this new model, we keep all constraints developed in Formulation 1, with the exception of constraints (12)-(14). The replacing constraints are the following ones:

$$w_{ij} = \sum_{p \in P} t_{ij}(p), \qquad \forall\ i, j \in I : i \neq j, \tag{30}$$

$$m \times (1 - x_{ij}) \geq w_{ij} - 1 \qquad \forall\ i, j \in I : i \neq j, \tag{31}$$

Constraints (30) translate the definition of the variables w_{ij} as explained above. Constraints (31) impose that if $w_{ij} \geq 2$, which means that if the distance between two haplotypes $i, j \in I$ is larger than 2, then $x_{ij} = 0$, i.e. that these haplotypes cannot be adjacent in the optimal phylogeny.

Regarding the SEC, it is important to mention that the implementation has to be different too. Here again, the constraints are implemented through a *break subtour procedure*, that checks if some haplotypes are isolated from the main tree. However, the constraint added to the model if it is not the case, is not the same as in the first two formulations. The new constraint imposes that the optimal solution must have either an edge between a haplotype belonging to the main tree and an isolated haplotype, or must consider an additional haplotype. Indeed, there is no guarantee that the graph of haplotypes the model built to form the phylogeny is connected, thus imposing a new edge could lead to a problem having no existent solution. As the model minimizes the number of haplotypes, the addition of this constraint will, in priority and if possible, break the subtour, and add a haplotype otherwise.

5 Experiments

In this section, we present and analyze the performance of our models in the solving of the problem stated in Section 3. We have conducted experiments in order to compare the different formulations we developed. Our main goal is to determine which of them is the most efficient in terms of *runtime*, *gap*, and *nodes*. The runtime is the time needed by the model to find the optimum, while the "gap" corresponds to the difference between the value of the optimum found and the value of root relaxation, divided by the optimum (Catanzaro et al., 2010). It is calculated as follows:

$$Gap(\%) = \frac{f^* - f_{LP\ relaxation}}{f^*} * 100 \tag{32}$$

and is expressed in percentages. Finally, the nodes are the number of nodes that are explored in the search tree until the optimum is found. We also intend to study the relevance of alternative formulations and evaluate the necessity of respecting the definition of adjacency as it has been formulated above.

Original size	Post-reduction			
	Min	Average	Max	Number of instances
5×5	1×2	$2 \times 3,35$	4×5	17
5×10	2×4	$2,91 \times 5,36$	4×7	11
10×5	$x \times x$ [2]	$x \times x$	$x \times x$	19
10×10	$x \times x$	$x \times x$	$x \times x$	5

Table 1: Summary of instances used for numerical experiments.

It is worth mentioning that the problem presented here has not been addressed yet in the literature, and therefore it is not possible to compare the performance of our formulations with the one achieved in other researches. One may argue that we could compare our results with the ones of the successive executions of Catanzaro et al. (2010) model for PPH problem and Catanzaro et al. (2013b) model for MPPEP, but this would not be relevant, as the comparison measures would not be practically comparable. The reasons are simple: it does not make sense to compute the gap, or the number of nodes explored of two successive models; and Catanzaro et al. (2013b) model for MPPEP has not been run on a data set that could correspond to the output of the PPH, making it impossible to compare them to our results, unless we implement them ourselves. For this reason, we will only compare the three formulations developed in Section 4.

Finally, it is important to understand that the objective of this master's thesis is not to advocate in favor of the Maximum Parsimony criterion over the other criteria. Nor do we claim that our model has the capacity to infer haplotypes, or to find the real phylogeny that relates these haplotypes. The idea here is only to assess the capacity of our formulations to address the discrete optimization problem stated above.

5.1 Implementation

We implemented Formulations 1, 2 and 3 by means of FICO Xpress Mosel 64-bit v5.0.0, Optimizer version 36.01.03, running on a MacPro 28-Core, 2.5GHz Intel Xeon W, 28 cores, 56 threads, 66.5MB cache, 196Gb 2933MHz and operating system MacOS X Catalina. We activated Xpress Optimizer automatic cuts and the pre-solving strategies during the runtime of the formulations. Finally, we have used the Xpress-MP primal heuristic to generate the first upper bound to each instance considered.

5.2 Data sets

The data sets we used to conduct our experiments have been extracted from a single data set, which is a biological sample of chromosome 10. It was initially used in Brown and Harrower (2006) and Catanzaro et al. (2010), and contains 36 genotypes made up of 30 SNPs each. We conducted some preliminary tests to choose sample sizes that are realistic in terms of runtime and relevant for a performance analysis. Thus, we extracted data sets of four different sizes: 5×5, 5×10, 10×5 and 10×10 from this biological sample. For each size, we generated 20 samples (except for the instances of size 10×10, for which we generated only 5 samples, due to timing and technical problems), so we had a total of 65 of them. The extraction strategy was the following: we picked randomly a site on a genotype, and then we selected as sample a table having the size chosen, where the picked site was the first element of the first row.

This strategy, despite its ease of use, suffers from a drawback. To understand it, it is necessary to specify that the genotypes of the input set are not directly processed by the model: they are subject to a pre-sorting. First, as explained above, our models are only made for genotypes having heterozygous sites, as they do not consider the possibility that a genotype would be resolved by two identical haplotypes. This involves to remove from the sample the genotypes made up solely of 0 and 1. It also removes the duplicate entries, in order to keep only distinct genotypes. Finally, among the remaining genotypes, it deletes identical columns in order to simplify the computation, so that if two sites are the same for each genotype, it removes the second SNP from each genotype. An example of this pre-sorting process can be found in Figure 3. The consequence of this sorting is that the data set processed by the model is almost never the size that was initially chosen, or is even sometimes reduced to an empty sample and is therefore not processed. Table 1 summarizes the size and the number of samples that are effectively processed by the model.

[2] Due to technical problems related to lockdown, this data has not been saved and is not available at the time of submission of this master's thesis.

Step 0 – Initial data set

Genotypes	SNP				
Genotype 1	**0**	**1**	**1**	**0**	**1**
Genotype 2	2	1	0	1	0
Genotype 3	2	1	0	1	0
Genotype 4	0	0	2	0	1
Genotype 5	0	2	1	2	1

Step 1 – Removing of genotypes having no heterozygous site (Genotype 1)

Genotypes	SNP				
Genotype 2	**2**	**1**	**0**	**1**	**0**
Genotype 3	**2**	**1**	**0**	**1**	**0**
Genotype 4	0	0	2	0	1
Genotype 5	0	2	1	2	1

Step 2 – Removing of duplicate entries (Genotypes 2 and 3 were identical)

Genotypes	SNP				
Genotype 2	2	**1**	0	**1**	0
Genotype 4	0	**0**	2	**0**	1
Genotype 5	0	**2**	1	**2**	1

Step 3 – Removing of duplicate entries of SNPs (the 2nd and 4th sites were identical for each genotype)

Genotypes	SNP			
Genotype 2	2	1	0	0
Genotype 4	0	0	2	1
Genotype 5	0	2	1	1

Figure 3: Example of the pre-sorting process on an input of genotypes.

5.3 Numerical results

Preliminary tests allowed us to notice that Formulation 3, although it is built on an interesting approach, is much less efficient than the first two, especially in terms of runtime. The total time required to run the preliminary test instances with this formulation is about 6 times longer than the time required with Formulations 1 and 2. We can explain this weakness by the size of the set I, which grows exponentially with the number of SNPs, and therefore makes the calculations much more cumbersome. For this reason, and in order to reduce the total running time, we only experimented further on Formulations 1 and 2. Thus, for the execution of both formulations on the different samples we generated, we used three performance indicators: runtime, gap and number of nodes explored.

Regarding the runtime and the number of nodes explored in the search tree, we analyzed the results as follows: we built *box and whisker plots* to compare the performances of both models on each data set. Box and whisker plots provide useful information: they represent on a graph the minimum, 1st quartile, median, 3rd quartile and maximum values of a set. Usually, we consider that if the median value of a model is comprised between the 1st and 3rd quartile of its rival model, we cannot conclude that it is better than the other on this performance measure.

It is worth mentioning that this comparison tool is particularly useful in situations where the values are fairly homogeneous. In our case, since the samples are reduced in a pre-sorting process, the different instances of a data set do not all have the same cardinality. This creates outliers in the results, that must be removed from the observations analyzed in the box and whisker plots, so that they are readable. Therefore, we got rid of a few instances: one in the 5×5 data set, three in the 5×10 data set and three in the 10×5 data set. However, we do not mean that these outliers are insignificant, so we should not ignore them in the analysis. Thus, any conclusions drawn from the graphs of the reduced results must be supported by the calculations made on the full results.

The box and whisker plots of runtime of Formulations 1 and 2, with instances of size 5×5, 5×10, 10×5 and 10×10, can be respectively found in Figures 4, 5, 6 and 7: neither model seems to particularly outperform the other in terms of runtime. The only one that could make us doubt this assertion is represented in Figure 7. However, despite appearances, the median of the runtime obtained by formulation 2 is lower than the 3rd quartile of formulation 1, since these values are respectively 7157,014 seconds and 7204,072 seconds. Moreover, it is necessary to underline the fact that because this data set is smaller than the others (it contains only 5 samples), it is more difficult to generalize the observations made from these results.

The full results (i.e. the ones comprising the outliers) support these findings. The median values obtained

Size	Model	Min	1st quartile	Median	3rd quartile	Max
5×5	Formulation 1	0,001	0,002	0,263	1,16	300,068
	Formulation 2	0,001	0,002	0,22	1,102	525,526
5×10	Formulation 1	0,15	0,2665	2,493	10,9165	1923,572
	Formulation 2	0,185	0,2605	2,604	151,526	15915,222
10×5	Formulation 1	0,002	0,229	2,385	6,94	4941,905
	Formulation 2	0,002	0,2425	2,351	6,0975	7175,335
10×10	Formulation 1	1,049	1,055	945,084	7204,072	20978,05
	Formulation 2	0,917	0,96	7157,014	14405,463	14473,177

Table 2: Details of computational analysis of the runtime.

Size	Model	Min	1st quartile	Median	3rd quartile	Max
5×5	Formulation 1	1	1	83	5347	1130972
	Formulation 2	1	1	47	2785	395429
5×10	Formulation 1	11	48	15853	65824	1829421
	Formulation 2	1	60	26641	710733,5	2163654
10×5	Formulation 1	1	18	4663	38453	335804
	Formulation 2	1	36	8797	46091	250239
10×10	Formulation 1	3159	3159	20276	91077	313763
	Formulation 2	1691	1691	60376	83764	131586

Table 3: Details of computational analysis of the number of nodes explored.

on the two different models are very close to each other, and also inside the "box" of the other model. The details of these values can be found in Table 2.

The box and whisker plots of the number of nodes explored by Formulations 1 and 2 with instances of size 5×5, 5×10, 10×5 and 10×10 can be respectively found in Figures 8, 9, 10 and 11. Here again, it seems that no model shows better performance than the other one. If Formulation 1 has median values slightly lower than the medians of Formulation 2, the difference is not sufficient to draw conclusions. As shown in Table 3, this trend reverses in some cases when the complete data are analyzed.

Although the runtime has already ruled out the Formulation 3, these first two performance measures do not allow us to rule out one of the two formulations we are studying. Then, the gap is identical in both models for each instance. Its value depends on two elements: the value of the objective function optimized, and the value of the same function with the integrity constraints relaxed. Without limit of time, both formulations compute the same value for the optimum; and for some reason, the value of the relaxation is worth 2 in any case. In light of these different elements, the two formulations seem very similar. To go further in the analysis, we can compare their constraints in order to determine if one of them has a polygon of possible values smaller than the other, which would make it more restrictive and therefore a better formulation.

The main difference between the two formulations lies in variables y, namely y_{ij}^k in Formulation 1 and y_i^k in Formulation 2. In particular, the following relationship holds:

$$y_i^k = \sum_{j \in I : j > i} y_{ij}^k + \sum_{j \in I : j < i} y_{ji}^k \qquad \forall\ k \in K,\ i \in I. \tag{33}$$

To begin with, it is worth comparing models through the constraints (5) in Formulation 1, that are replaced by (23) in Formulation 2. Starting from (23), we note that

$$\sum_{i \in I} y_i^k = \sum_{i \in I} \Big(\sum_{j \in I : j > i} y_{ij}^k + \sum_{j \in I : j < i} y_{ji}^k \Big) = \Big(\sum_{i \in I} \sum_{j \in I : j > i} y_{ij}^k \Big) + \Big(\sum_{i \in I} \sum_{j \in I : j < i} y_{ji}^k \Big) = 2 \qquad \forall\ k \in K. \tag{34}$$

Writing this is therefore equivalent to the sum of two constraints (5). We can therefore already conclude that Formulation 2 is a surrogate relaxation of Formulation 1, and is therefore less powerful. We can go a little further by comparing the constraints (3) and (4) in Formulation 1, which are replaced in Formulation 2 by the constraints (22). They can be rewritten as follows:

$$y_i^k \leq u_i \iff \sum_{j \in I : j > i} y_{ij}^k + \sum_{j \in I : j < i} y_{ji}^k \leq u_i \qquad \forall\ k \in K,\ i \in I. \tag{35}$$

Then,

$$y_{ij}^k \leq \sum_{q \in I: q > i} y_{iq}^k + \sum_{q \in I: q < i} y_{qi}^k \leq u_i \qquad \forall\ k \in K,\ i, j \in I : i \neq j, \tag{36}$$

shows that Formulation 1 is here also more restrictive, as u_i, which must be minimized, has a lower bound than in Formulation 2.

This analysis shows that Formulations 1 and 2 are very similar, but nevertheless present some differences, that allow us to state that Formulation 1 outperforms Formulation 2. In that event, we recommend to use Formulation 1 to solve the problem posed above. Regarding Formulation 3, whose performance is more limited than the ones of the two first formulations, it can be interesting in some situations as it solves a more demanding problem. Since it offers features that it is the only one to, we should not completely rule it out.

6 Conclusion

In this master's thesis, we investigate a particular variant of the haplotyping problem, subject to a tree constraint, under Maximum Parsimony. We propose several formulations, based on Catanzaro et al. (2010) and Catanzaro et al. (2013b). The idea behind our formulations is to minimize the total size of the phylogeny that contains the haplotypes necessary to solve a set of genotypes. The rules for constructing the phylogeny depend on the formulation, in some cases allowing to connect two haplotypes in the tree regardless of the distance between them, in other cases allowing a link only if the distance between them is equal to 1.

Computational experiments showed that, although the formulations we propose provide exact solutions for instances of the stated problem, their performance is quite limited. Judging the absolute performance by the computing time would not be representative since it depends on the computer used. Thus, we will discuss the absolute performance of our formulations in terms of gap. As briefly mentioned above, the value of the linear relaxation is equal to 2 for Formulations 1 and 2, regardless of the input. On the other hand, the optimal value of the objective function increases with the number of genotypes to solve. Consequently, the higher the number of genotypes to be resolved, the larger the gap will be. In the case of a single genotype, the value of the optimum is at least 3 (2 haplotypes, connected by an edge of weight 1), so the gap cannot be smaller than $\frac{3-2}{3} = 33.333\%$. Regarding Formulation 3, the value of the linear relaxation is equal to 3 in any case, making its gap slightly better, but still poor. We therefore recommend using these models on instances of limited size, especially with a limited number of SNPs for the use of Formulation 3.

Despite their questionable performance, our models are interesting because they offer a solution to a problem that has not been addressed so far. In future research, it would be interesting to compare their results with those obtained by successive use of a haplotyping model and a phylogeny construction model. If the outcome of these tests is positive, our models could be considered as a real contribution to the literature, and could be adapted for use in other practical applications. In future researches, it may also be interesting to explore different possibilities of implementation of the break subtour procedure, for which there is room for improvement.

Acknowledgements

In writing this master's thesis, I was supported by several people whom I would like to thank. First of all, I would like to express my gratitude to Daniele Catanzaro, my promoter, for his continuous support, his knowledge, his motivation, his patience. I would also like to thank my father, Ella Mae Hall and Victoria Tichy, who agreed to reread my work in details. Finally, I would like to thank my parents, family and friends for all the support they gave me during my studies.

References

B. Atri and O. Lichtarge. Computational approaches to studying molecular phylogenetics. In A. Shanker, editor, *Bioinformatics: Sequences, Structures, Phylogeny*, chapter 9, pages 173–190. Springer, Singapore, 2018.

C. Bleidorn. *Phylogenomics: An Introduction.* Springer, Cham, 2017.

G. E. Blelloch, K. Dhamdhere, E. Halperin, R. Ravi, R. Schwartz, and S. Sridhar. Fixed parameter tractability of binary near-perfect phylogenetic tree reconstruction. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 667–678. Springer, 2006.

A. BW. Boldt, I. J. Messias-Reason, D. Meyer, C. G. Schrago, F. Lang, B. Lell, K. Dietz, P. G. Kremsner, M. L. Petzl-Erler, and J. FJ. Kun. Phylogenetic nomenclature and evolution of mannose-binding lectin (mbl2) haplotypes. *BMC Genetics*, 11, 2010.

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

S.K. Burley, S. C. Almo, J. B. Bonanno, M. Capel, M. R Chance, T. Gaasterland, D. Lin, A. Sali, F. W. Studier, and S. Swaminathan. Structural genomics: beyond the human genome project. *Nature*, 23: 151–157, 1999.

D. Catanzaro. Estimating phylogenies from molecular data. In R. Bruni, editor, *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, pages 146–176. Springer, New York, 2010.

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

D. Catanzaro, A. Godi, and M. Labbé. A class representative model for pure parsimony haplotyping. *INFORMS Journal on Computing*, 22:195–209, 2010.

D. Catanzaro, M. Labbé, and B. V. Halldórsson. An integer programming formulation of the parsimonious loss of heterozygosity problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10, 2013a.

D. Catanzaro, R. Ravi, and R. Schwartz. A mixed integer linear programming model to reconstruct phylogenies from single nucleotide polymorphism haplotypes under the maximum parsimony criterion. *Algorithms for Molecular Biology*, 8, 2013b.

F. Collins and D. Galas. A new five-year plan for the u.s. human genome project. *Science*, 262:43–46, 1993.

F. S. Collins and V. A. McKusick. Implications of the human genome project for medical science. *JAMA*, 285:540–544, 2001.

F. S. Collins, A. Patrinos, E. Jordan, A. Chakravarti, R. Gesteland, and L. Walters. New goals for the u.s. human genome project: 1998-2003. *Science*, 282:682–689, 1998.

D. C. Crawford and D. A. Nickerson. Definition and clinical importance of haplotypes. *Annual Review of Medicine*, 56:303–320, 2005.

R. W. DeBry. The consistency of several phylogeny-inference methods under varying evolutionary rates. *Molecular Biology and Evolution*, 9:537–551, 1992.

H. C. Fan, J. Wang, A. Potanina, and S. R. Quake. Whole-genome molecular haplotyping of single cells. *Nature*, 29:51–57, 2011.

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

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

G. CL. Johnson, L. Esposito, B. J. Barratt, A. N. Smith, J. Heward, G. DiGenova, H. Ueda, H. J. Cordell, I. A. Eaves, F. Dudbridge, R. CJ. Twells, F. Payne, W. Hughes, S. Nutland, H. Stevens, P. Carr, E. Tuomilehto-Wolf, J. Tuomilehto, S. CL. Gough, D. G. Clayton, and T. A. Todd. Haplotype tagging for the identification of common disease genes. *Nature*, 29:233–237, 2001.

J. C. Long, A. Chakravarti, C. D. Boehm, S. Antonarakis, and H. H. Kazazian. Phylogeny of human β-globin haplotypes and its implications for recent human evolution. *American Journal of Physical Anthropology*, 8:113–130, 1990.

National Human Genome Institute. The human genome project, 2019. URL `https://www.genome.gov/human-genome-project`.

T. Niu, Z. S. Qin, X. Xu, and J. S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Science Direct*, 70, 2002.

E. Paradis. *Analysis of Phylogenetics and Evolution with R*. Springer, New York, 2012.

A. Rzhetsky and M. Nei. Statistical properties of the ordinary least-squares, generalized least-squares, and minimum-evolution methods of phylogenetic inference. *Journal of Molecular Evolution*, 35:367–375, 1992.

R. Schwartz. Computational models for cancer phylogenetics. In T. Warnow, editor, *Bioinformatics and Phylogenetics*, volume 29 of *Computational Biology*, chapter 11, pages 243–275. Springer, 2019.

M. Steel and D. Penny. Parsimony, likelihood, and the role of models in molecular phylogenetics. *Molecular Biology and Evolution*, 17:839–850, 2000.

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

The International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409:928–933, 2001.

T. Warnow. Large-scale multiple sequence alignment and phylogeny estimation. In C. Chauve, N. El-Mabrouk, and E. Tannier, editors, *Models and Algorithms for Genome Evolution*, Computational Biology, chapter 6, pages 85–146. Springer, 2013.

T. Warnow. Divide-and-conquer tree estimation: Opportunities and challenges. In T. Warnow, editor, *Bioinformatics and Phylogenetics*, volume 29 of *Computational Biology*, chapter 6, pages 121–150. Springer, Cham, 2019.

I. J. Wilson, M. E. Weale, and D. J. Balding. Inferences from DNA data: Population histories, evolutionary processes and forensic match probabilities. *Journal of the Royal Statistical Society*, 166:155–188, 2003.

Z. Yang and B. Rannala. Molecular phylogenetics: principles and practice. *Nature*, 13:303–314, 2012.

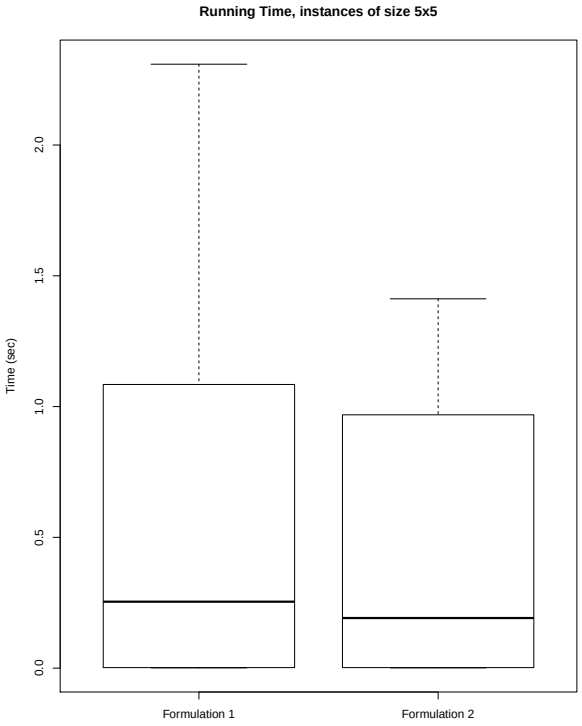

Figure 4: Box and whisker plot of runtime, 5×5 data set.

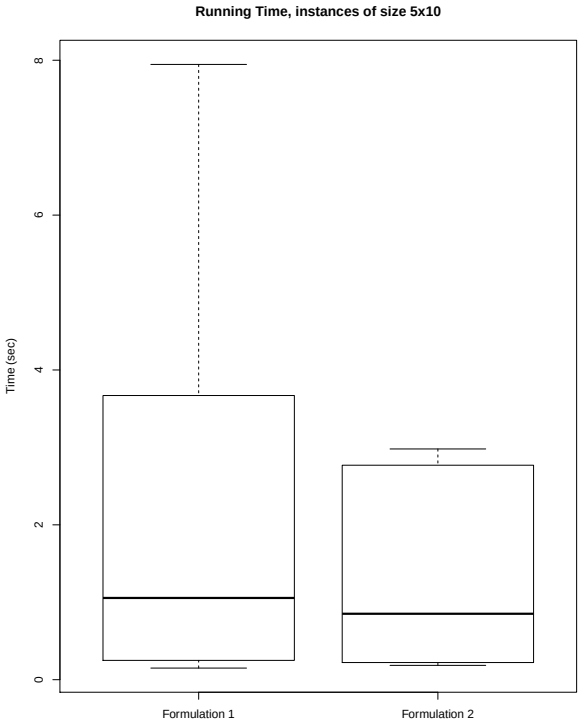

Figure 5: Box and whisker plot of runtime, 5×10 data set.

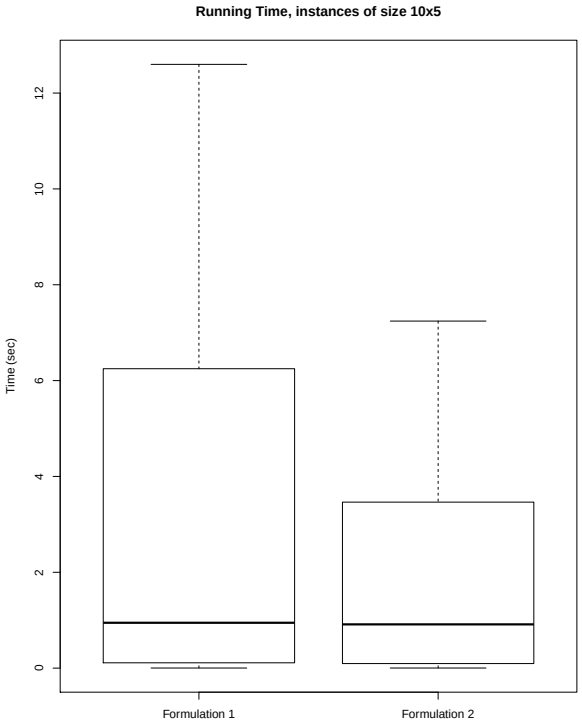

Figure 6: Box and whisker plot of runtime, 10×5 data set.

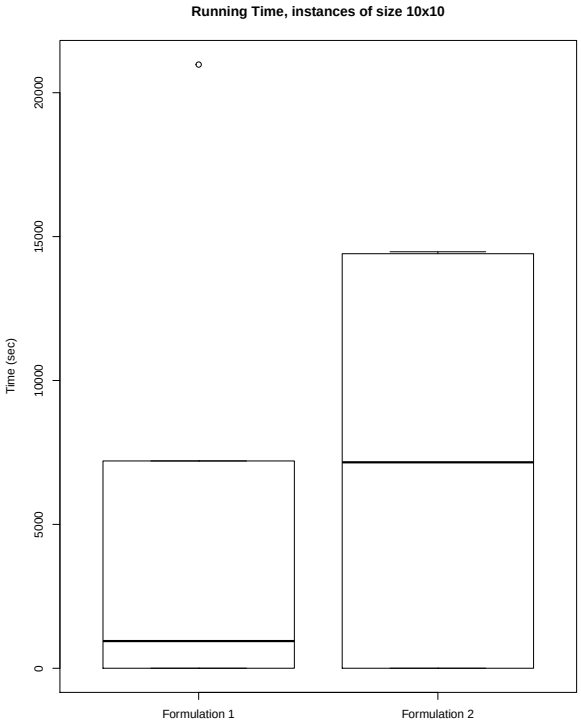

Figure 7: Box and whisker plot of runtime, 10×10 data set.

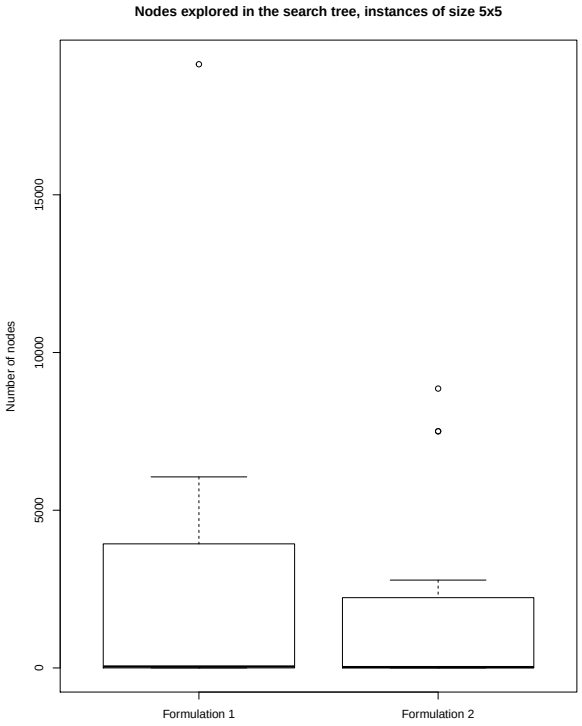

Figure 8: Box and whisker plot of number of nodes, 5×5 data set.

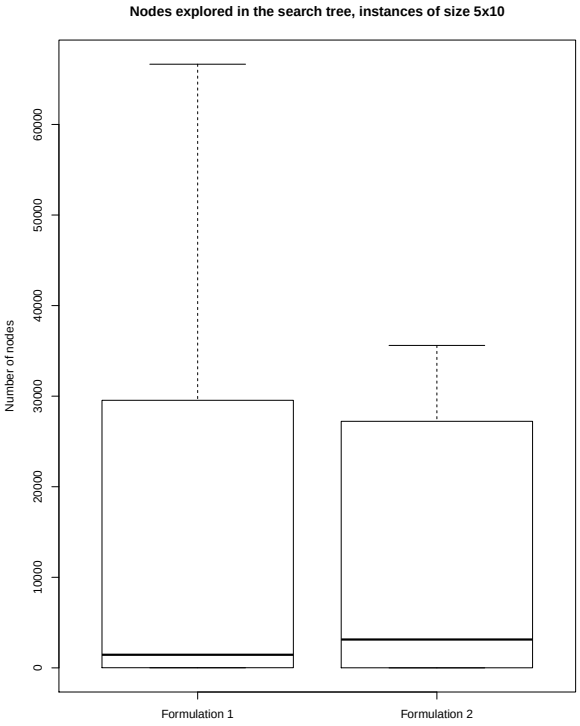

Figure 9: Box and whisker plot of number of nodes, 5×10 data set.

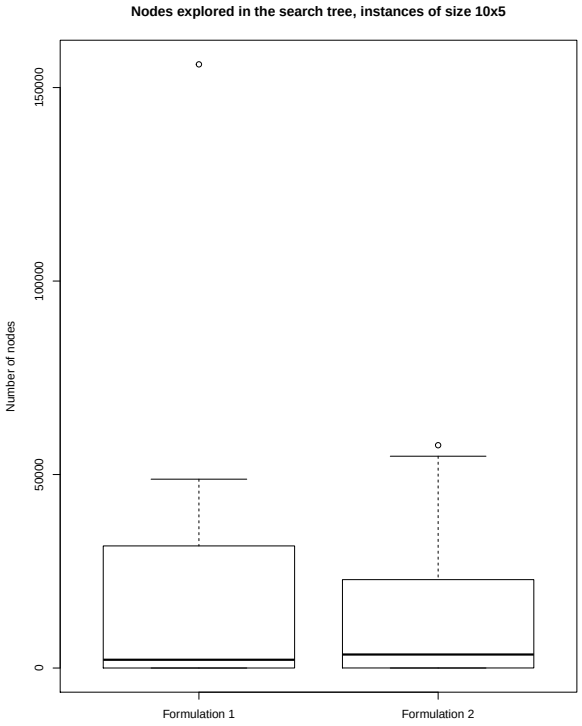

Figure 10: Box and whisker plot of number of nodes, 10×5 data set.

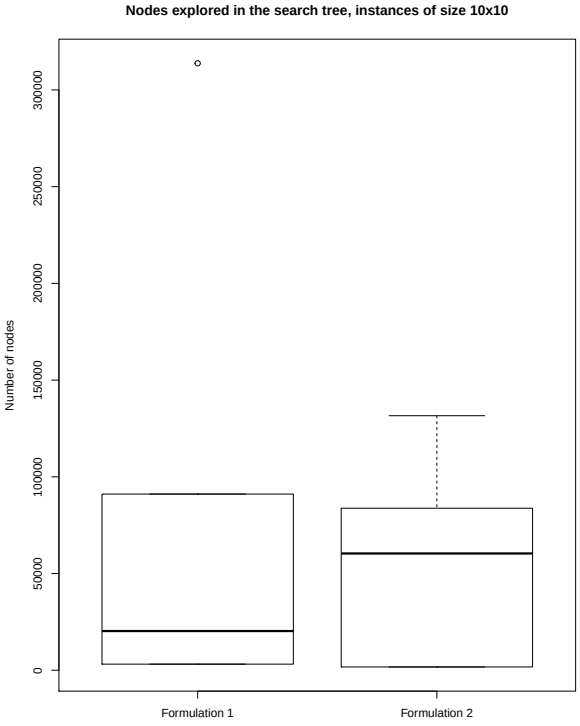

Figure 11: Box and whisker plot of number of nodes, 10×10 data set.

LOUVAIN-LA-NEUVE | BRUXELLES | MONS | TOURNAI | CHARLEROI | NAMUR
Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/lsm**