9-11-2006

# Algorithms for Computational Genetics Epidemiology

Jingwu He
jingwu@cs.gsu.edu

# ALGORITHMS FOR COMPUTATIONAL GENETIC EPIDEMIOLOGY

by

Jingwu He

Under the Direction of Alex Zelikovsky

## ABSTRACT

The most intriguing problems in genetics epidemiology are to predict genetic disease susceptibility and to associate single nucleotide polymorphisms (SNPs) with diseases. In such these studies, it is necessary to resolve the ambiguities in genetic data. The primary obstacle for ambiguity resolution is that the physical methods for separating two haplotypes from an individual genotype (phasing) are too expensive. Although computational haplotype inference is a well-explored problem, high error rates continue to deteriorate association accuracy. Secondly, it is essential to use a small subset of informative SNPs (tag SNPs) accurately representing the rest of the SNPs (tagging). Tagging can achieve budget savings by genotyping only a limited number of SNPs and computationally inferring all other SNPs. Recent successes in high throughput genotyping technologies drastically increase the length of available SNP sequences. This elevates importance of informative SNP selection for compaction of huge genetic data in order to make feasible fine genotype analysis. Finally, even if complete and accurate data is available, it is unclear if common statistical methods can determine the susceptibility of complex diseases.

The dissertation explores above computational problems with a variety of methods, including linear algebra, graph theory, linear programming, and greedy methods. The contributions include (1)significant speed-up of popular phasing tools without compromising their quality,  (2)stat-of-the-art tagging tools applied to disease association, and (3)graph-based method for disease tagging and predicting disease susceptibility.

INDEX WORDS: Tagging, Phasing, Haplotype, Genotype, SNP,
Disease association, Susceptibility prediction

# ALGORITHMS FOR COMPUTATIONAL GENETIC EPIDEMIOLOGY

by

Jingwu He

A Dissertation Submitted in Partial Fulfillment of Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2006

ALGORITHMS FOR COMPUTATIONAL GENETIC EPIDEMIOLOGY

by

Jingwu He

| | |
|---|---|
| Major Professor: | Alex Zelikovsky |
| Committee: | Yi Pan |
| | Anu Bourgeois |
| | Ion Mandoiu |

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
December 2006

# DEDICATION

*To my dear daughter, Jennifer, my wife, Jun and my parents*

# ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Alexander Zelikovsky for advising and guide for my Ph.D dissertation. Secondly, I want to thank my dissertation committee members, Dr. Yi Pan, Dr. Anu Bourgeois and Dr. Ion Mandoiu. I also appreciate support and assistance from our research group: Dumitru Brinza, Kelly Westbrooks, Weidong Mao and Nisar Hundewale. Finally, I want to thank my family and friends for their support and beliefs.

# TABLE OF CONTENTS

Page

CHAPTER

# LIST OF TABLES

**Table** **Page**

# LIST OF FIGURES

**Figure** **Page**

# CHAPTER 1

# INTRODUCTION

Recent improvement in accessibility of high-throughput DNA sequencing brought a great deal of attention to disease association and susceptibility studies. Successful genome-wide searches for disease-associated gene variations have been recently reported [52, 86]. However, complex diseases can be caused by combinations of several unlinked gene variations. This proposal addresses computational challenges of discovering causal gene combinations and accurate predicting susceptibility to common complex diseases. The number of typed *single nucleotide polymorphisms* (SNPs) for disease association and linkage studies is reaching 250,000 from SNP Mapping Arrays [1]. High density maps of SNPs as well as massive DNA data with large number of individuals and number of SNPs become publicly available [5]. It is a computational challenge to analyze and data-mine such huge volumes. This dissertation meets this challenge to develop corresponding highly scalable computational tools.

In diploid organisms each chromosome has two "copies" which are not completely identical. Each of two single copies is called a haplotype, while a description of the data consisting of mixture of the two *haplotypes* is called a *genotype*. For complex diseases caused by more than a single gene it is important to obtain haplotype data which identify a set of gene alleles inherited together. In haplotype description it is important only positions where the two copies are different which are called *single nucleotide polymorphisms* (SNPs). A SNP is a single nucleotide site where exactly two (of four) different nucleotides occur in a large percentage of the population. Biologists only consider those variation occurring at least 1% of population as SNPs (see Figure

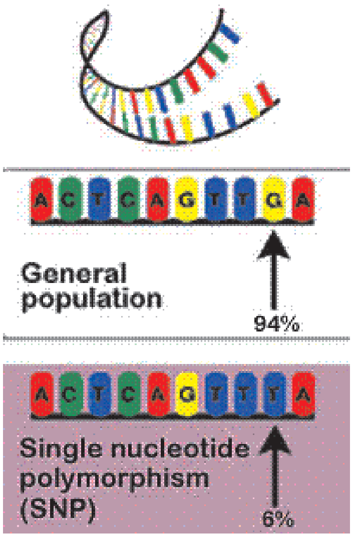


**Figure 1.1.** SNPs

1.1. In total, there exits 10 million SNPs in human population. The SNP-based approach for disease association study is the dominant one, and high density SNP maps have been constructed across the human genome with a density of about one SNP per thousand nucleotides.

In general, it is costly and time consuming to examine the two copies of a chromosome separately, and genotype data rather than haplotype data are only available, even though it is the haplotype data that will be of greatest use. Data from $m$ sites (SNPs) in $n$ individual genotype are collected, where each site can have one of two states (alleles). For each individual, we would ideally like to describe the states of the m sites on each of the two chromosome copies separately, i.e., the haplotype. However, experimentally determining the haplotype pair is technically difficult or expensive. Instead, the screen will learn the 2m states (the genotype) possessed by the individual, without learning the two desired haplotypes for that individual. One then uses computation to infer haplotype information from the given genotype information, called haplotype inference problem (or phasing problem). Several methods have been explored and some are intensively used for this task [20, 21, 32, 34, 65, 69, 75]. None of these methods are presently fully satisfactory, although many give impressively accurate results. Chapter 3 of this dissertation devotes to this task. In Section 3.1, we speeds up popular haplotype inference tools while finding almost the same solution practically in all cases thus not compromising the quality of the known haplotype inference methods. For the perfect phylogeny reconstruction we reduce the runtime by factor of 60. In Section 3.2, we propose two new greedy and integer linear programming for phasing family trios, and extensive experimental validation of proposed methods showing advantage over the previously known methods.

The search for the association between complex diseases and single nucleotide polymorphisms (SNPs) has been recently received great attention. For these studies, it is essential to use a small subset of informative SNPs, named *tags*, accurately representing the rest of the SNPs. Firstly, informative SNPs can be used for selective SNP typing and computationally inferring all non-typed SNPs thus achieving considerable budget savings. Secondly, informative SNPs can be used for compaction of SNP data. Indeed, recent successes in high throughput genotyping technologies (e.g., Affimetrix Map Arrays) drastically increase the length of available SNP sequences and they should be compacted to be feasible for fine genotype analysis. Chapter 4 of this dissertation proposes stat-of-the-art informative SNP seleciton tools for applying to disease association study.

The main goal of disease susceptibility analysis is to identify gene variations or, in general, haplotypes and genotypes which are susceptible to a particular disease. If complex diseases are affected by multiple genes, the traditional direct statistical association so far is unsatisfactory and arguably is not applicable since it mostly relies on an assumption that the disease is caused by a single Mendelian gene [22], but some complex diseases, such as psychiatric disorders, are characterized by a non mendelian, multifactorial genetic contribution with a number of susceptible genes interacting with each other[11, 68]. Statistical association analysis usually results in claims that a presence of a given SNP considerably increases the risk of a certain disease which are of limited use for disease susceptibility because of the following two reasons. Firstly, it is difficult to derive a meaningful conclusion in case of a disease probability being, e.g., 10 in a million and the resulted increased probability being 20 in a million - such a negligible absolute probability increase is unreliable. Secondly, the SNPs susceptibility to complex diseases are usually linked and do not, therefore, have an increased cumulative impact as it would be expected from the independent SNPs. The observed weakness of statistical methods may lead to a quite plausible assertion that each case of complex diseases may have a unique chain of genetic as well as environmental elements [22]. Chapter 5 of this dissertation explores possibility of applying combinatorial methods to known case/control studies with the hope to reliably (to certain extent) predict disease susceptibility.

## 1.1 Road Map and Contributions

In Section 3.1, we propose a new linear algebra based method for speeding up popular software tools (e.g. PHASE[87], HAPLOTYPER[69] and DPPH[24] for haplotype inference, since those tools are often not well scalable. When the number of sites (SNPs) comes to thousands these tools often cannot deliver answer in reasonable time even if the number of haplotypes is small. The new linear algebra based method drastically reduces the number of sites in the original data. After solving a reduced instance, linear decoding allows to recover haplotypes of full length for given genotypes. Experiments show that our method significantly speeds up popular haplotype inference tools while finding almost the same solution practically in all cases thus not compromising the quality of the known haplotype inference methods. For the perfect phylogeny reconstruction we reduce the runtime by factor of 60.

In Section 3.2, we propose two new greedy and integer linear programming for phasing family trio data which are commonly obtained in disease association study.

Genotype data represent family trios consisting of the two parents and their child since that allows to recover haplotypes with higher confidence. Although there exist many phasing methods for unrelated adults or pedigrees, phasing and missing data recovery for trios is lagging behind. We have tried several well-known computational methods for phasing Daly et al. [26] family trio data, but, surprisingly, all of them give infeasible solutions with high inconsistency rate. We formally propose two new greedy and integer linear programming based solution methods, and extensive experimental validation of proposed methods showing advantage over the previously known methods.

In Section 4.1, we describe previous work on informative SNP selection and formulate the problem. In Section 4.2, we propose linear algebraic methods for solving the problem. This method is based on Gauss-Jordan elimination that is used to predict non-tag SNP by rounding fractional linear combination over tag SNPs. We obtain an extremely good compression and prediction rates. For example, for long haplotypes ($> 25000$ SNPs), knowing only 0.4% of all SNPs we predict the entire unknown haplotype with 2% accuracy while the prediction method is based on a 10% sample of the population.

In Section 4.3, we show how to separate the tag selection from SNP prediction, formulate the corresponding optimization problem, and describe the general approach and two heuristics for tag selection based on prediction.

In Section 4.4, we proposes a new SNP prediction method based on multiple linear regression (MLR) analysis in sigma-restricted coding. When predicting a non-tag SNP, the MLR method accumulates information about all tag SNPs resulting in significantly higher prediction accuracy with the same number of tags than for the previously known tagging methods. We also show that the tag selection strongly depends on how the chosen tags will be used – advantage of one tag set over another can only be considered with respect to a certain prediction method. Two simple universal tag selection methods have been applied: a (faster) stepwise and a (slower) local-minimization tag selection algorithms. An extensive experimental study on various datasets including 10 regions from HapMap shows that the MLR prediction combined with stepwise tag selection uses significantly fewer tags (e.g., up to two times less tags to reach 90% prediction accuracy) than the state-of-art methods of Halperin et al. [41] for genotypes and Halldorsson et al. [42] for haplotypes, respectively. Our stepwise tagging matches the quality of while being faster than STAMPA [41].

In Section 4.5, we proposes a new SNP prediction using a robust tool for classification – Support Vector Machine (SVM). For tag selection we use a fast stepwise tag selection algorithm. An extensive experimental study on various datasets including three regions from HapMap shows that the tag selection based on SVM SNP prediction can reach the same prediction accuracy as the methods of Halldorson et al. [42] on the LPL using significantly fewer tags. For example, our method reaches 90% non-tag SNP prediction accuracy using only three tags for Daly et al. [26] dataset with 103 SNPs. The proposed tagging method is also more accurate (but considerably slower) than multiple linear regression method of He et al. [46].

In Section 4.6, we use MLR tagging [46] to reduce set of SNPs. we propose to apply a novel combinatorial method for finding disease-associated multi-SNP combinations. Our experimental study shows that the proposed methods are able to find multi-SNP combinations whose disease association is statistically significant even after multiple testing adjustment. For Daly et al [26] data we found a few unphased multi-SNP combinations associated with Crohn's disease with multiple testing adjusted p-value below 0.05 while no single SNP or pair of SNPs show any significant association. For Ueda et al [93] data we found a few new unphased and phased multi-SNP combinations associated with autoimmune disorder.

In Chapter 5, we first propose a greedy set covering method for removing irrelevant SNPs but still keeping disease information, and then describe several prediction algorithms which are mostly based on combinatorial optimization. We apply proposed methods to two data sets. The first data set consists of case/control study of Crohn's disease [26] of 129 family trios. The other set for autoimmune disorder [93] consists of 1036 unrelated case/control individuals. We achieved correct prediction rate of 77.28% and 64.77%, respectively. After applying bootstrapping we obtain with 95% confidence the correct prediction rate of 75.38% for Crohn's disease. We have also performed a *Monte-Carlo test* by running our methods on Crohn's disease's data with randomly swapped case/control markers. The average prediction rate falls to 50% for all proposed methods. This confirms predominating genetic susceptibility of Crohn's disease [7], high association of the chosen haplotype region with Crohn's disease as well as capabilities of the proposed methods to detect such susceptibility.

# CHAPTER 2

# BIOLOGY BACKGROUND: SNPS, HAPLOTYPES, GENOTYPES, AND NOTATIONS

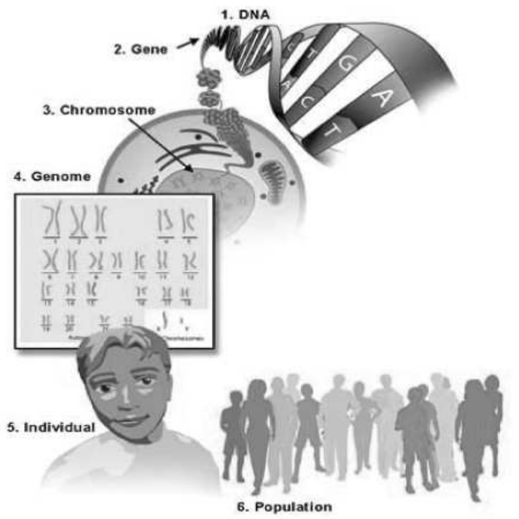


**Figure 2.1.** DNA, gene, chromosome, genome

Usually all living organisms are organized in 4 levels: Genome, chromosomes, genes, and DNA (see Figure 2.1). DNA is a double helical molecule with specific base pairing rules. Each of the two strands of the double helical structure serves as a template for synthesis of a new DNA strand during replication. Before a cell divides, the DNA within the cell nucleus is copied with exceptional fidelity. Information in DNA is organized into Genes, which is the second level. Genes make up Chromosomes, and all chromosomes taken together form an organism's Genome. Every cell in an Individual contains the genome. Cells are the fundamental working units of every living organism. Each cell contains a complete copy of an organism's genome. The genome is distributed along chromosomes, which are made of compressed and entwined DNA.

A gene is a segment of chromosomal DNA that directs the synthesis of a protein. DNA is made of two complimentary strands of nucleotides. A's complement is T and G's complement is C. Usually the more the living organism has evolved, the longer genome they have. The length of DNA is measured by the number of base pairs (bp).

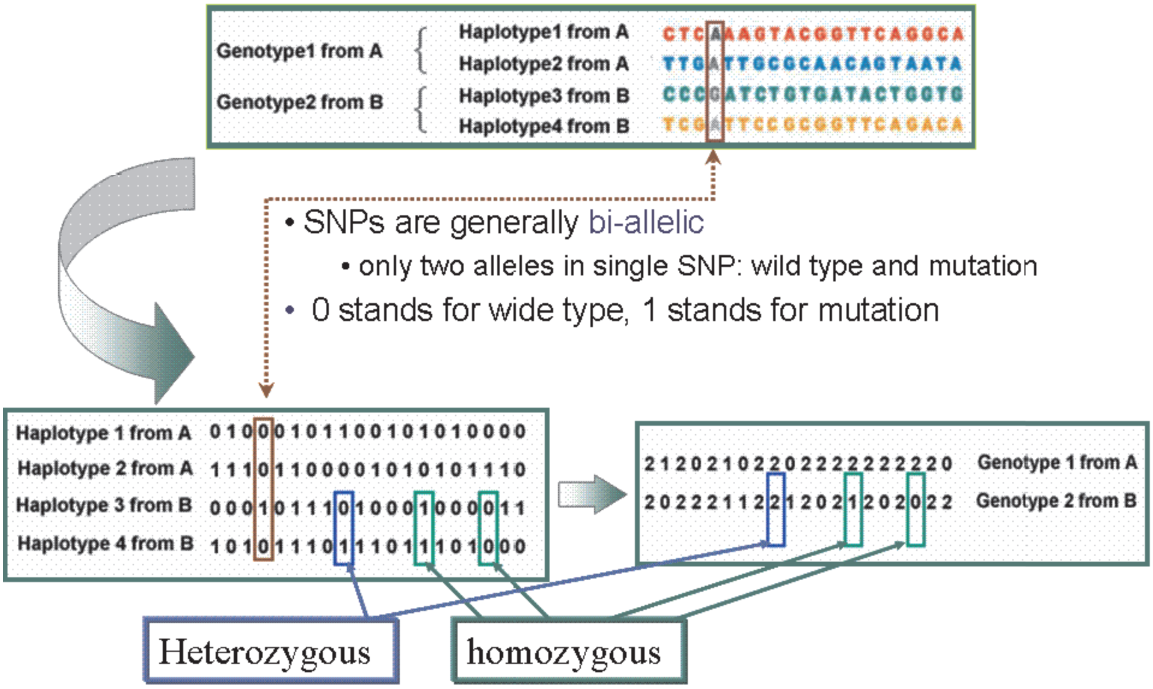
Humans have 46 total chromosomes, two copies of each of 23 different types. Chromosomes 1 through 22 are the same in both males and females. The sex (X and Y) chromosomes differ between the sexes. Males have one X and one Y chromosome, whereas females have two X and no Y chromosomes. One copy of each chromosome type is inherited from the mother and one from the father. A father contributes an X chromosome to each of his daughters and a Y chromosome to each of his sons.

In diploid organisms each chromosome has two "copies" which are not completely identical. Each of two single copies is called a haplotype, while a description of the data consisting of mixture of the two *haplotypes* is called a *genotype*. For complex diseases caused by more than a single gene it is important to obtain haplotype data which identify a set of gene alleles inherited together. Genome difference between any two people is about 0.1% of genome. These differences are Single Nucleotide Polymorphisms (SNPs). Both substitutions have to be observed in the general population at a frequency greater than 1%. SNP's occur as frequently as every 100-300 bases. This implies that in an entire human genome there are approximately 10 to 30 million potential SNP's. More than 4 million SNP's have been identified and the information has been made publicly available. SNPs may occur in both coding (gene) and non-coding regions of the genome. Many SNPs have no effect on cell function, but they could predispose people to disease or influence their response to a drug.

The differences between any two human individuals are produced by mutation, crossing over and genetic recombination during fertilization (union of egg and sperm). Mutation is the change in DNA of an organism which may result in that organism being different than its parents. While there are many causes of mutations, some factors are known which rapidly increase the incidence of mutation. In crossing over which occurs in the production of sex cells or gametes in meiosis, there is an exchange of chromosome pieces between the chromosome pairs associated with each other in this process

SNP's are bi-allelic and can be referred as 0 if it's a majority and 1, otherwise. If both haplotypes are the same allele, then the corresponding genotype is homogeneous, can be represented as 0 or 1. If the two haplotypes are different, then the genotype is represented as 2 (See Figure 2.2. Usually the major allele is expected to be the wild type and the minor allele is expected to be a mutation. It is important to study SNPs because they represent genetic differences among humans. Therefore biologists are searching for risk factors for genetic diseases among SNPs.

The Human Genome Project [5] is the organized, international effort to map and sequence the entire human genome. Much information about the human genome including maps and sequences are available through the internet. The great majority of the human DNA sequence has now been determined.


Genotype1 from A
Haplotype1 from A CTCAAAGTACGGTTCAGGCA
Haplotype2 from A TTGATTGCGCAACAGTAATA
Genotype2 from B
Haplotype3 from B CCCGATCTGTGATACTGGTG
Haplotype4 from B TCGATTCCGCGGTTCAGACA
Haplotype 1 from A 0 1 0 0 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 0
Haplotype 2 from A 1 1 1 0 1 1 0 0 0 0 1 0 1 0 1 0 1 1 1 0
Haplotype 3 from B 0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 0 0 0 1 1
Haplotype 4 from B 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 0 0
2 1 2 0 2 1 0 2 2 0 2 2 2 2 2 2 2 2 2 0 Genotype 1 from A
2 0 2 2 2 1 1 2 2 1 2 0 2 1 2 0 2 0 2 2 Genotype 2 from B
Heterozygous
homozygous

- SNPs are generally bi-allelic
  - only two alleles in single SNP: wild type and mutation
- 0 stands for wide type, 1 stands for mutation

**Figure 2.2.** Encode

# CHAPTER 3

# HAPLOTYPE INFERENCE PROBLEM

In general, it is costly and time consuming to examine the two copies of a chromosome separately, and genotype data rather than haplotype data are only available, even though it is the haplotype data that will be of greatest use. One then uses computation to extract haplotype information from the given genotype information. Several methods have been explored and some are intensively used for this task [20, 21, 32, 34, 65, 69, 75]. None of these methods are presently fully satisfactory, although many give impressively accurate results. In section 3.1, we propose a new linear algebra based method which drastically reduces the number of sites in the original data. After solving a reduced instance, linear decoding allows to recover haplotypes of full length for given genotypes. Experiments show that our method significantly speeds up popular haplotype inference tools while finding almost the same solution practically in all cases thus not compromising the quality of the known haplotype inference methods. For the perfect phylogeny reconstruction we reduce the runtime by factor of 60.

In disease association study, family trio data are commonly obtained. genotype data represent family trios consisting of the two parents and their child since that allows to recover haplotypes with higher confidence. Although there exist many phasing methods for unrelated adults or pedigrees, phasing and missing data recovery for trios is lagging behind. We have tried several well-known computational methods for phasing Daly et al. [26] family trio data, but, surprisingly, all of them give infeasible solutions with high inconsistency rate. In section 3.2, we propose two new greedy and integer linear programming based solution methods, and extensive experimental validation of proposed methods showing advantage over the previously known methods.

## 3.1 Population Haplotype Inference Problem

### 3.1.1 Previous Work and Problem Formulation

In diploid organisms each chromosome has two “copies” which are not completely identical. Each of two single copies is called a haplotype, while a description of the data consisting of mixture of the two haplotypes is called a genotype. For complex diseases caused by more than a single gene it is important to obtain haplotype data which identify a set of gene alleles inherited together. In haplotype description it is important only positions where the two copies are different which are called single nucleotide polymorphisms (SNPs). A SNP is a single nucleotide site where exactly two (of four) different nucleotides occur in a large percentage of the population. The SNP-based approach is the dominant one, and high density SNP maps have been constructed across the human genome with a density of about one SNP per thousand nucleotides.

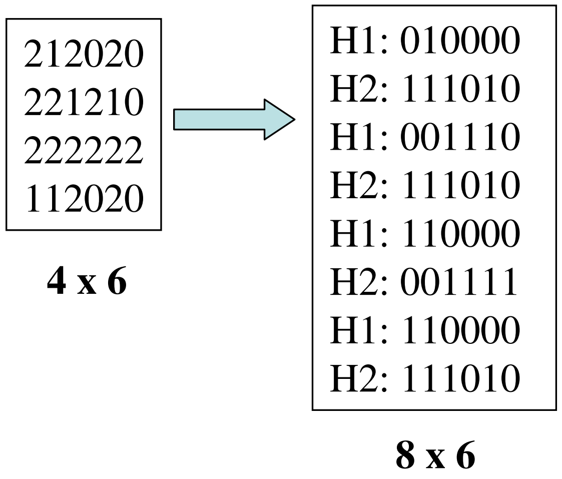
In general, it is costly and time consuming to examine the two copies of a chromosome separately, and genotype data rather than haplotype data are only available, even though it is the haplotype data that will be of greatest use. Data from $m$ sites (SNPs) in $n$ individual genotype are collected, where each site can have one of two states (alleles), which we denote by 0 and 1. For each individual, we would ideally like to describe the states of the m sites on each of the two chromosome copies separately, i.e., the haplotype. However, experimentally determining the haplotype pair is technically difficult or expensive. Instead, the screen will learn the 2m states (the genotype) possessed by the individual, without learning the two desired haplotypes for that individual. One then uses computation to extract haplotype information from the given genotype information.

Population haplotype inference problem asks for a set of haplotypes explaining a given set of genotypes. The input and the output of the Haplotype Inference problem admits the following traditional combinatorial description (see e.g., [25]).

The input population is given in the form of an $n \times m$ *genotype matrix* $G = \{g_{ij}\}$ with all values $g_{ij} \in \{0, 1, 2\}$. Each row $g_i$, $i = 1, \ldots, n$, of the matrix $G$ corresponds to a genotype and each column $s_j$, $j = 1, \ldots, m$, corresponds to a site of interest on the chromosome, namely, a SNP. When the site $s_j$ is homozygous for the genotype $g_i$, then $g_{ij} = 0$ if the associated chromosome site has that state 0 on both copies and, respectively, $g_{ij} = 1$ if the site has state 1 on both copies. When the site $s_j$ is heterogenous for the genotype $g_i$, i.e., the site has different state on the two copies, then $g_{ij} = 2$.

The output of Haplotype Inference problem is a $2n \times m$ *haplotype matrix* $H = \{h_{ij}\}$, with all values $h_{ij} \in \{0, 1\}$. A consecutive pair of rows $(h_{2i-1}, h_{2i})$ corresponds to a pair of haplotypes which is a feasible "explanation" of the genotype vector $g_i$, $i = 1, \ldots, n$. For any homozygous site $s_j$ of the genotype $g_i$, i.e., the site with value 0 (respectively, 1), the corresponding haplotypes should both have value 0 (respectively, 1) in its $j$-th position, i.e., if $g_{ij} = 0$, then $h_{2i-1,j} = h_{2i,j} = 0$ and if $g_{ij} = 1$, then $h_{2i-1,j} = h_{2i,j} = 1$. For any heterogenous site $s_j$ of the genotype $g_i$, i.e., the site with value 2, the corresponding haplotypes should have different values in its $j$-th position, i.e., if $g_{ij} = 2$, then $h_{2i-1,j} = 1 - h_{2i,j}$. We can see an example of Haplotype Inference Problem as Figure 3.1

Thus, the Haplotype Inference problem asks for a haplotype matrix $H$ which is a feasible "explanation" of a given genotype matrix $G$. Although the input and the output of the Haplotype Inference problem are very well formalized, it is still ill-formulated since, in general as well as in common biological setting, there is ex-



212020
221210
222222
112020

4 x 6

H1: 010000
H2: 111010
H1: 001110
H2: 111010
H1: 110000
H2: 001111
H1: 110000
H2: 111010

8 x 6

**Figure 3.1.** An example of Haplotype Inference Problem

ponential number of possible haplotype matrices for the same input matrix. Indeed, an individual genotype with $k$ heterozygous sites can have $2^{k-1}$ haplotype pairs that could appear in $H$. Without additional biological insight, one cannot deduce which of the exponential number of solutions is the best, i.e., the most biologically meaningful.

A variety of methods have been developed to solve the HI problem. There are two major approaches to solving the inference problem: combinatorial methods and statistical methods. Combinatorial methods often state an explicit objective function that one tries to optimize in order to obtain a solution to the inference problem. Statistical methods are usually based on an explicit model of haplotype evolution; the inference problem is then cast as a maximum-likelihood or a Bayesian inference problem. The most widely used algorithm in combinatorial methods is Clark's Algorithm Clark [21], and expectation-maximization (EM) algorithm is the most important statistical method [69, 87].

Clark et al. [21] introduced a program, called HAPINFERX. The algorithm begins by listing all possible haplotypes that must be present unambiguously in the sample. This list comes from those individuals whose haplotypes are unambiguous from their genotypes, that is, those individuals who are homozygous at every locus.

If no such individuals exist, then the algorithm cannot start (at least, not without extra information or manual intervention). Once this list of known haplotypes has been constructed, the haplotypes on this list are considered one at a time, to see whether any of the unresolved genotypes can be resolved into a known haplotype plus a complementary haplotype. Such a genotype is considered resolved, and the complementary haplotype is added to the list of known haplotypes. The algorithm continues cycling through the list until all genotypes are resolved or no further genotypes can be resolved in this way. The solution obtained can (and often does) depend on the order in which the genotypes are entered.

Stephens et al. [87] introduced a Bayesian statistical method PHASE for phasing genotype data. It exploits ideas from population genetics and coalescent theory that make phased haplotypes to be expected in natural populations. It also estimates the uncertainty associated with each phasing. The software can deal with SNP in any combination, any size of population and missing data are allowed. The drawback of this method is that it takes long time for large population

Niu et al. [69] proposed a new Monte Carlo approach HAPLOTYPER for phasing genotype data. It first partition the whole haplotype into smaller segments then use the Gibbs sampler both to construct the partial haplotypes of each segment and to assemble all the segments together. This method can accurately and rapidly infer haplotypes for a large number of linked SNPs. The drawback of HAPLOTYPER is that it can not handle lengthy genotype with large population. It limits 100 SNPs and 500 population.

Brinza *et al.*[14] introduced a scalable phasing method: 2SNP. In this method, haplotypes for each genotype are inferred based on the maximum spanning tree of a complete graph with vertices corresponding to heterozygous sites. The edge weights of the genotype graph express the confidence (based on linkage disequilibrium and distance between SNPs) in the most probable phasing of 2-SNP genotypes. The computation of edge weights takes in account statistically significant deviations from expected 2-SNP genotype phasing and from the random mating model. 2SNP is extremely fast comparatively with probabilistic EM algorithms, its runtime is $O(nm(n+m))$, where $n$ and $m$ are the numbers of genotypes and SNPs, respectively. As a result, it can solve very large instances of the phasing problem.

The 2SNP algorithm is described in detail in Figure 3.2.

```
Input:  $n \times m$ genotype matrix $G = (g_{i,j}|\ g_{i,j} \in \{0,1,2,?\},\ i = 1..n,\ j = 1..m)$

For each pair of SNPs $i$ and $j$, $i = 1..m, i = 1..m$ do
  - Compute observed haplotype frequencies $F_{00}, F_{01}, F_{10}, F_{11}$
  - Estimate $P_{22}$ and $C_{22}$, the number of parallel and cross phasings of 22
    genotypes, adjusted to deviation from the random mating model
  - Compute the edge weight $w_{ij}$ by formula using haplotype
    frequencies adjusted with $P_{22}$ and $C_{22}$
For each genotype $g_i, i = 1..n$ do
  - Find maximum spanning tree of the weighted genotype graph $G(g_i)$
  - Color genotype graph $G(g_i)$ into two colors such that MST edges with
    positive weights will connect vertices with the same color and edges
    with negative weights will connect vertices with opposite colors
  - Phase genotype $g_i$ such that heterozygous SNPs are phased in parallel
    if corresponding vertices have the same color and cross if different.
For each haplotype recover ?'s according to the haplotype that is closest
with respect to Hamming distance.

Output: $2n \times m$ haplotype matrix $H = (h_{i,j}|\ h_{i,j} \in \{0,1\},\ i = 1..2n,\ j = 1..m)$
```

**Figure 3.2.** 2SNP Phasing Algorithm

### 3.1.2 Linear Dependence of Sites, Haplotypes and Genotypes

In this section we give motivation and informal description of ideas behind suggested linear reduction of haplotype inference methods.

Usually, in genetic sequences derived from human haplotypes (see [26, 78]), the number of sites is much larger than the number of individuals. Because of such disproportion many columns corresponding to SNP sites are similar. Indeed, as noted in [78], the number of *synonymous* sites in real data is considerably large, here two sites are synonymous (or equivalent) if the corresponding 0-1-columns either the same or the complimentary (i.e., the same after each entry $x$ is replaced with $1 - x$). It is common to keep only one site out of several synonymous sites since they are assumed not to carry any additional information [78]. Thus if the site column $s_i$ is equal or complementary to the site column $s_j$, then one of them can be dropped. From haplotype inference point of view, we infer the haplotypes in one of the synonymous sites the same way as in another.

we make the next inductive step: if $k$ columns are "dependent", or $k$-th site can be "expressed" in terms of $k-1$ others, then we suggest to drop the $k$-th site. Indeed, the $k$-th site arguably does not carry any information additional to one which we can derive from the first $k - 1$ sites. Inductively, if we decide how to infer haplotypes in the first $k - 1$ site, then we should consistently infer haplotypes in the $k$-th site.

In order to make this idea work, we need to formalize the notion of "dependent" or "expressed" in a such way that it should be easy and fast to derive and manipulate. The most suitable approach is to rely on the standard linear dependence. Unfortunately, two synonymous 0-1-columns are not linearly dependent in a standard arithmetic. As noted in [8], one cannot straightforwardly apply linear combinations of column-sites since *equivalent* columns are linearly independent. It is not difficult to see that replacing 0's with -1's will resolve that issue. Indeed, in the new notations, multiplication by (-1) corresponds to complementing the column in the traditional notations. Thus

**Remark 1** *In (-1,1)-notations, two sites are synonymous if and only if they are collinear (i.e., linearly dependent).*

We also need to change notations for genotypes. Ideally, a genotype obtained from two haplotypes should be linear dependent from these haplotypes, then we can hope that linear dependency between columns of the genotype matrix will correspond to linear dependency between columns of the haplotype matrix. It is easy to see that replacing 0's with -1's (as for haplotypes) and replacing 2's with 0's makes this idea work. In the new notations,

**Remark 2** *In (-1,1,0)-notations, a genotype vector $g$ is obtained from haplotype vectors $h$ and $h'$ if and only if $g = (h + h')/2$.*

One can also explore linear dependency of rows-haplotypes rather than columns-SNPs. Then linear dependency in $(-1, 1)$-notations can be used for classification of recombinations. Assume that in the given population all recombinations happen at a limited number of hotspots. Assume further that each hotspot occupies a DNA segment between two consecutive SNPs. If initially there are only two haplotypes $a$ and $b$, then by repeatedly recombining $a$ and $b$ at $g$ different hotspots, one can potentially obtain as much as $2^{g+1}$ different haplotypes.

Indeed, let $a = a_1 a_2 \ldots a_{g+1}$ and $b = b_1 b_2 \ldots b_{g+1}$, where $a_1$ (respectively, $b_1$) is the segment of $a$ (resp. $b$) from the first SNP to the last SNP before the first hotspot, $a_i$ (respectively, $b_i$) is the segment between $(i-1)$-st and $i$-th hotspots, and $a_{g+1}$ (respectively, $b_{g+1}$) is the segment from the last hotspot to the last SNP. Then any haplotype $h$ obtained by recombination of $a$ and $b$ can be partitioned into $g + 1$ segments each coming either from $a$ or from $b$, i.e., $h = h_1 \ldots h_{g+1}$ where $h_i = a_i$ or $h_i = b_i$.

On the other hand, the number of linearly independent recombinations of two haplotypes is at most $g + 2$ which is much smaller then $2^{g+1}$ which allows

**Theorem 3** *Let $\mathcal{H}$ be a set of haplotypes obtained from two haplotypes by recombination events at $g$ hotspots. Then the number of linearly independent rows-haplotypes is at most $g + 2$, i.e., the linear rank of $\mathcal{H}$, $rank(\mathcal{H}) \leq g + 2$.*

Let initial two haplotypes be $a$ and $b$, and let $g$ hotspots partition them into substrings as follows $a = a_1a_2 \dots a_{g+1}$ and $b = b_1b_2 \dots b_{g+1}$. Consider the set of $g + 2$ vectors which consists of the vector $a$ and vectors $b_i$ each having all substrings (except the $i$-th substring) equal 0 and the $i$-th substring equal $b_i - a_i$, i.e., $b_i = 0 \dots (b_i - a_i) \dots 0$, $i = 1, \dots, g + 1$. Any recombination haplotype vector $h = h_1h_2 \dots h_{g+1}$ can be represented as

$$h = a + \sum_{h_i = b_i} b_i$$

The proof of the following theorem is similar.

**Theorem 4** *Let $\mathcal{H}$ be a set of haplotypes obtained from $l$ different haplotypes by recombination events at $g$ hotspots. Then the number of linearly independent rows-haplotypes is at most $(g + 1)(l - 1) + 1$, i.e., the linear rank of $\mathcal{H}$, $rank(\mathcal{H}) \leq (g + 1)(l - 1) + 1$.*

Obviously, the number of linearly independent columns $r$ cannot be more than the size of population, i.e., the number of rows. Also, Remark 2 implies that $r$ is at most $h$, where $h$ is the number of haplotypes. In next section we explore how the linear reduction can reduce the runtime for all known haplotype inference methods.

### 3.1.3 Implementation of Linear Reduction Based on Matrix Multiplication

In this section we describe linear algebra behind the suggested implementation of our linear reduction. Everywhere further we will only use new (-1,1,0)-notations for genotypes and haplotypes.

Let $G$ be a (-1,1,0)-genotype matrix consisting of $n$ rows corresponding to genotypes and $m$ columns corresponding to SNP sites. We will modify the (-1,1)-haplotype matrix $H$ by removing all duplicate rows, i.e., if a haplotype is used for different genotypes, then only a single its copy remains in $H$. Let the modified matrix $H'$ has $h$ rows. The dependency between $G$ and $H'$ can be expressed as a graph $X = (H, G)$

with $h$ vertices corresponding to haplotypes and $n$ edges corresponding to genotypes – an edge connects two vertices if the corresponding genotype row is a sum of the corresponding two haplotype rows. Let $I_X$ be an $n \times h$ incidence matrix of the graph $X$, i.e., each of row $e_i$ of $I_X$ corresponds to a genotype $g_i$ and consists of all 0's except exactly two 1's in two columns corresponding to the two vertices-haplotypes connected by $e_i$. Thus, using matrix multiplication we can express this dependency as follows

$$G = I_X \times H' \tag{3.1}$$

One can reformulate the Haplotype Inference problem as follows: given a (-1,1,0)-matrix $G$, find a (-1,1)-matrix $H'$ and a graph $X$, such that the equality (3.1) holds. In other words, the Haplotype Inference problem becomes equivalent to a matrix factorization problem (3.1) (see Figure 3.3).

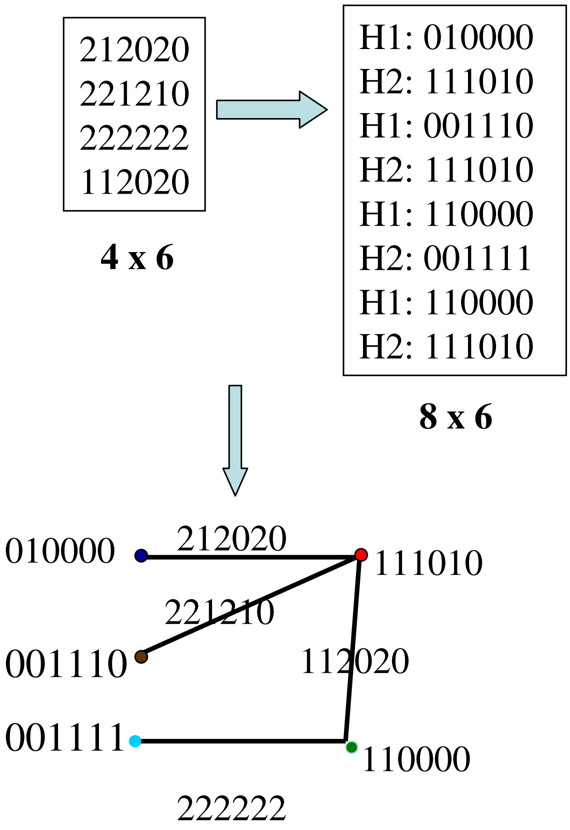


**Figure 3.3.** An graph representation of Haplotype Inference Problem

We apply linear reduction to haplotype inference using above new notations (-1, 1, 0). The proposed linear reduction consists of the following three steps:

1. (*encoding*) reduce the genotype matrix by keeping only linearly independent sites and dropping all linearly dependent sites;
2. apply an arbitrary haplotype inference method to the resulted site-reduced genotype matrix obtained;
3. (*decoding*) complement the inferred site-reduced haplotype matrix with linearly dependent column-sites which are obtained using original linear combinations of inferred haplotype columns.

Let $rh$ be the rank of the matrix $H'$. Note that the number of sites is often larger than the number of haplotypes, $m >> h$, therefore $rank(H')$ often coincide with the number of rows $h$. The matrix $H'$ can be represented as follows

$$H' = H_{rh} \times (E_{rh}|C) \tag{3.2}$$

where the matrix $H_{rh}$ consists of $rh$ linearly independent columns of $H'$ and $(E_{rh}|C)$ is a $(rh \times m)$ matrix with the first $rh$ columns forming the identity matrix $E_{rh}$ (1's on the main diagonal and 0's elsewhere) and $C$ is a $(rh \times (m - rh))$ matrix. Substituting (3.2) into (3.1), we obtain

$$G = I_X \times H_{rh} \times (E_{rh}|C) \tag{3.3}$$

On the other hand, using $O(n^2m)$ Gaussian elimination, we can extract $r = rank(G)$ linearly independent columns from the matrix $G$ such that

$$G = G_r \times (E_r|C') \tag{3.4}$$

where the matrix $G_r$ consists of $r$ linearly independent columns of $G$ and $(E_r|C')$ is a $(r \times m)$ matrix with the first $r$ columns forming the identity matrix $E_r$ and $C'$ is a $(r \times (m - r))$ matrix.

If the matrix $rank(I_X) = rh$ (note that $rank(I_X) \leq rh$), then $r = rh$. If we can choose the same linearly independent sites for $G$ and $H$, then (3.3) and (3.4) implies that $C = C'$ and

$$G_r = I_X \times H_{rh} \tag{3.5}$$

Thus, we have reduced the Haplotype Inference problem (3.1) to the *linearly reduced* Haplotype Inference problem (3.5). Indeed, in time $O(n^2m)$ we find representation (3.4), then after solving factorization (3.5), we can find $H'$ using (3.2) in time $O(h^2m)$. For example, the PPH problem can be solved in time $O(n^2m)$ rather than in time $O(nm^2)$.

### 3.1.4 Fixing Caveats in Linear Reduction Approach

Unfortunately, the plan, mentioned in previous section, may fail since the factorization problem (3.5) may have more solutions than original problem (3.1). It is possible that the matrix $H'$ obtained from (3.2) contains entries not equal to -1 and 1 or, even worse, there is no feasible matrix $H'$ which can be obtained from $H_{rh}$. This section we show how to enhance the original linear reduction idea to deal with this two caveats.

In our experiments we have found that sometimes the matrix multiplication

$$H_{rh} \times (E_r|C') \tag{3.6}$$

which presumably gives the decoded complete haplotype matrix $H'$, results in non-feasible product with entries unequal to -1 and 1. This happens when the matrix $I_X$ does not have full column rank, i.e., when $rank(I_X)$ is less than the number of columns of $I_X$. The following theorem characterizes *nontrivial* graphs, i.e., graphs which have full column rank incidence matrix $I_X$.

**Theorem 5** *The graph $X$ is nontrivial if and only if each connected component of $X$ is not bipartite.*

Let the graph $X = (V, E)$ have a bipartite connected component with vertex set $C$, i.e., $C = C_A \cup C_B$ and all edges with at least one endpoint in $C$ should connect a vertex from $C_A$ with a vertex from $C_B$. Then the sum of all columns corresponding to $C_A$ equals to the sum of all columns corresponding to $C_B$ since whenever $C_A$-column has 1 in a certain row, the 1 corresponding to the opposite endpoint of the edge will be in $C_B$ columns in the same row.

It is easy to see that $X$ is nontrivial if and only if each connected component is nontrivial. Wlog assume that $X$ is a connected graph with an odd cycle $B$. By removing edges we can assume that $X$ is a connected graph with a single odd cycle. If $X$ does not have leaves, then $X = B$ and by sorting vertices of $X$ in order of the traversal of $B$, we obtain that $I_X$ is a square matrix with $det(I_X) = 2$. If $X$ has a leaf $l$, then by induction $I_{X-l}$ has full column rank and $r(I_X) = r(I_{X-l}) + 1$ since the column $l$ has a single 1 in a new row.

The following lemmna shows why one would prefer the graph $X$ is nontrivial.

**Lemma 6** *Let $X$ be a trivial graph. Then there exists such a genotype matrix $G$ and two different haplotype matrix $H_1$ and $H_2$ such that $G = I_X \times H_1$ and $G = I_X \times H_2$.*

If $X$ is a trivial graph, then there exists a connected component $X'$ which can be colored into 2 colors. It is easy to see that we can find a matrix $H$ with a column $C$ having all 0's in the rows corresponding to the genotypes-edges from $X'$. We can color all vertices of $X'$ into two colors $(-1)$ and 1 such that no two adjacent vertices will have the same color. Obviously, such phase assignment corresponding to $H_1$

**Input:** The reduced genotype matrix $G = \{s_{ij}\}$, the graph $X_r$ and the haplotype matrix $H_r$
**Output:** The graph $X_m$ and the haplotype matrix $H'$

```
For each site $s_j$ and each connected component $C$ of $X_r$ do
  (1) Find an edge $e_i$ with with non-zero label $s_{ij}$
  (2) Label one endpoint $u$ of $e_i$ with $s_{ij}$, i.e., $s_{u,j} = s_{ij}$
  (3) In breadth-first-search manner, propagate the labels over $C$:
        for each edge $e_i = (u, v)$, $s_{v,j} = s_{u,j} - s_{ij}$
Output haplotype matrix $H'$ which are the labels of the vertices of the graph $X_m$.
```

**Figure 3.4.** The Decoding Algorithm.

is feasible as well as "opposite" phase assignment $H_2 = -H_1$ obtained from $H_1$ by multiplying each phase assignment in $X_1$ by $(-1)$.

Note that the probability that $X$ is nontrivial is very high when the number of edges (genotypes) is large enough with respect to the number of haplotypes. Nevertheless, we propose a simple decoding algorithm which reconstructs $H'$ from $H_{rh}$ which relies on the reconstruction of the graph $X$ rather than on the matrix multiplication.

As soon as the haplotype graph $X_r$ for the reduced set of sites of size $r$ is obtained by any haplotype inferring algorithm, we need to extend this graph to the graph $X_m$ for all $m$ sites. Very often the graphs $X_r$ and $X_m$ are isomorphic, i.e., they are the same graphs but with the different labels – $X_r$-vertices (resp. edges) are labeled by haplotypes (resp. genotypes) with the reduced set of $r$ sites while $X_m$-vertices (resp. edges) are labeled by haplotypes (resp. genotypes) with the required site-length of $m$. The following Decoding Algorithm (see Figure 3.4) will always restore $X_m$ from $X_r$ if $X_m$ and $X_r$ are isomorphic.

Note that the Decoding Algorithm would stuck at the step (1) if all edges of a connected components have a zero label for a certain site. But in this case, as we know from Remark 6, the solution is not unique if the connected component is bipartite and no feasible labels are possible otherwise. Therefore we have proved the following

**Theorem 7** *If the graph $X_m$ for original Haplotype Inference problem and the graph $X_r$ for the site-reduced Haplotype inference problem are isomorphic, then the decoding algorithm correctly end with the unique solution.*

In some cases the resulted matrix $H$ has invalid values, i.e., not equal to $(-1)$ or 1. Then, as we discussed before, the columns with invalid values belong to bipartite connected components. If there is at least one non-0 edge in a bipartite connected component $C$ in this column, then there is a unique way to correctly assign phases. Indeed, the phases of endpoints of a non-0 edge are uniquely determined (1's if this is 1-edge or $(-1)$, otherwise). All other vertices in the connected component will obtain a unique possible phase since the phase of one endpoint and the value of edge uniquely define the phase of the other endpoint.

In general, finding complete haplotype matrix $H$ from the reduced matrix $H'$ can be done using the method above rather than matrix multiplication. We only need to prove that a contradiction never happens.

Unfortunately, it is possible that the graphs $X_r$ and $X_m$ are not isomorphic. Indeed, consider the following two genotypes with three sites:

$$g_1 = (1, 0, 1) \text{ and } g_2 = (0, -1, -1) \tag{3.7}$$

The reduced site set includes the first two sites since the third column-site just equals the sum of the first two column-sites. The corresponding reduced haplotype graph $X_r$ has 3 vertices labeled $h_1 = (1, 1)$, $h_2 = (1, -1)$ and $h_3 = (-1, -1)$ and edges $g_1 = (h_1, h_2)$ and $g_2 = (h_2, h_3)$ (see Figure 3.5(a)), while $X_m$ has 4 vertices (see Figure 3.5(b)).

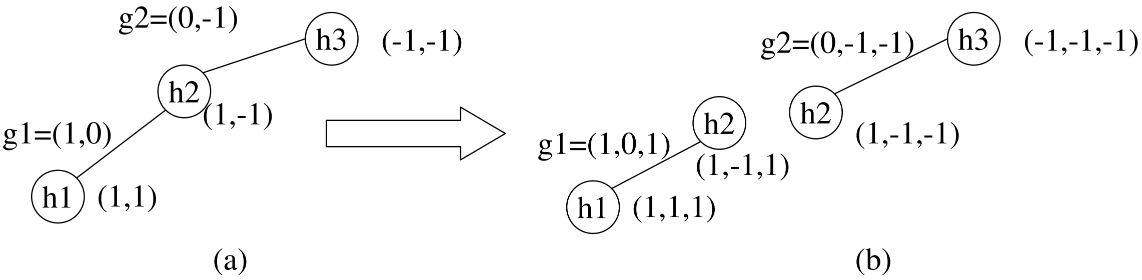


**Figure 3.5.** (a) The reduced haplotype graph with 3 vertices. (b) Result of splitting of the vertex $h_2$ into two vertices ).

Thus, if the graphs $X_m$ and $X_r$ are not isomorphic, then we should apply splitting of vertices. Fortunately, in our extensive experimental study we never got an instance where splitting was necessary.

### 3.1.5 Experimental Results

In generating the test data, following [25] we have used the haplotype generator ms [54]. This generator is a well-known standard based on the coalescent model of SNP sequence evolution. The ms generator has capability to generate a given number of haplotypes with the prescribed number of sites and recombination rate. In our tests, we have generated $2n$ haplotypes according to $n$ sites, and then randomly paired them to obtain $n$ genotypes. For the Table 3.1 we have set the recombination rate to 0. For all other testcases the recombination rate is specified in the corresponding tables. Although in principle, the linearly reduced haplotype inference methods may need to split vertices (see Section 3.5), in *all* our testcases it is never happened.

Three different methods of perfect phylogeny reconstruction (DPPH [24], GPPH [38], and HPPH [25]) have been compared in [25]. The experimental data show that the fastest method DPPH is slowly increases advantage in runtime over the second best HPPH achieving factor of 3 for the largest instances. In Table 3.1, using similar testcases we compare DPPH with the suggested Linearly Reduced DPPH. For the

| Datasets | | Average Running Time (seconds) | | | |
|---|---|---|---|---|---|
| | | DPPH | Linearly Reduced DPPH | | |
| sites | pop | Total | Total | E and D | RD |
| 30 | 50 | 0.0297 | 0.0141 | 0.0042 | 0.0099 |
| 100 | 100 | 0.2168 | 0.0793 | 0.0446 | 0.0374 |
| 300 | 150 | 1.9243 | 0.2144 | 0.1499 | 0.0645 |
| 500 | 250 | 7.9553 | 0.5825 | 0.4548 | 0.1277 |
| 1000 | 500 | 56.4723 | 2.6373 | 2.2311 | 0.4062 |
| 2000 | 1000 | 549.415 | 9.4355 | 8.0125 | 1.4230 |

**Table 3.1.** The comparison of the running times of DPPH and Linearly Reduced DPPH. Each value is averaged over 100 datasets. E and D is the CPU time for encoding and decoding and RD is DPPH runtime for the reduced instance.

Linearly Reduced DPPH we report separately the total CPU time as well as time taken by linear encoding and decoding and time taken by DPPH to solve the reduced instance. Our results show that the advantage in runtime of Linearly Reduced DPPH grows fast with testcase size and reaches factor of 60 for the largest instances. In *all* testcases, if DPPH finds a unique solution for the Haplotype Inference problem, then so does the Linearly Reduced DPPH and the solution is identical.

In Tables 3.2 and 3.4 we compare the runtime and quality of haplotype inference of the standard PHASE [87] and proposed Linearly Reduced PHASE. As we can see, the runtime is drastically reduced while the quality measured in the average number of errors is not significantly larger.

Similarly, the Table 3.5 and 3.7 we compare the runtime and quality of haplotype inference of the standard HAPLOTYPER [69] and proposed Linearly Reduced HAPLOTYPER.

The last two tables are devoted to biological datasets. The first three testcases are derived from the drosophila haplotypes and last two testcases are derived from the 616 kilobase region from [26].

| Datasets | | Average Running Time (seconds) | | | |
|---|---|---|---|---|---|
| | | PHASE | Linearly Reduced PHASE | | |
| sites | pop | Total | Total | E and D | RP |
| 60 | 30 | 201.9131 | 42.2205 | 0.1093 | 42.2124 |
| 60 | 60 | 501.2166 | 148.2856 | 0.0262 | 148.2594 |
| 100 | 50 | 1019.1806 | 162.4863 | 0.0392 | 162.4471 |
| 100 | 100 | 2103.3299 | 274.7445 | 0.0722 | 274.6722 |
| 140 | 70 | 2367.4557 | 141.2005 | 0.0606 | 141.1399 |
| 140 | 140 | 4460.5307 | 179.8020 | 0.1012 | 179.7008 |

**Table 3.2.** The comparison of the running times of PHASE and Linearly Reduced PHASE. Each value is averaged over 25 datasets.

| Datasets | | Recombination Rate | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | | | 4 | | | 16 | | | 40 | | |
| sites | pop | LRP/P | LRP/O | P/O | LRP/P | LRP/O | P/O | LRP/P | LRP/O | P/O | LRP/P | LRP/O | P/O |
| 60 | 30 | 6.25 | 9.91 | 8.84 | 10.53 | 14.25 | 11.98 | 19.42 | 20.07 | 12.80 | 20.08 | 17.55 | 15.92 |
| 60 | 60 | 6.47 | 9.28 | 7.99 | 10.76 | 8.88 | 10.15 | 8.05 | 9.86 | 6.62 | 10.72 | 12.68 | 9.19 |
| 100 | 50 | 9.02 | 12.92 | 11.46 | 9.00 | 10.24 | 8.88 | 12.84 | 14.81 | 11.28 | 22.38 | 22.61 | 10.36 |
| 100 | 100 | 5.88 | 8.39 | 8.46 | 6.01 | 8.14 | 7.63 | 5.66 | 8.78 | 5.07 | 7.34 | 9.38 | 6.64 |
| 140 | 70 | 8.39 | 9.38 | 10.02 | 8.59 | 8.91 | 9.84 | 9.32 | 6.25 | 8.48 | 12.92 | 11.60 | 10.49 |
| 140 | 140 | 6.29 | 9.66 | 11.11 | 7.97 | 10.18 | 9.82 | 9.28 | 11.17 | 10.54 | 12.65 | 14.37 | 11.56 |

**Table 3.3.** The comparison of the quality of haplotyping of Linearly Reduced PHASE (LRP) and PHASE (P) vs the original haplotypes (O). Here the difference in haplotype data sets, Hapset1/Hapset2 is the arithmetic mean of numbers of false-positive and false-negative haplotypes over the number of haplotypes Hapset2 times 100%. Each value is averaged over 25 datasets.

| Datasets | | Recombination Rate | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | | | 4 | | |
| sites | pop | LRP/P | LRP/O | P/O | LRP/P | LRP/O | P/O |
| 60 | 30 | 1.28 | 1.91 | 2.23 | 1.18 | 4.71 | 4.71 |
| 60 | 60 | 2.68 | 4.46 | 3.57 | 7.10 | 5.41 | 7.88 |
| 100 | 50 | 3.71 | 3.71 | 5.56 | 4.34 | 6.08 | 5.21 |
| 100 | 100 | 4.17 | 6.70 | 7.29 | 4.57 | 5.32 | 6.01 |
| 140 | 70 | 5.92 | 7.77 | 8.51 | 6.12 | 5.33 | 7.45 |
| 140 | 140 | 5.05 | 8.19 | 10.50 | 4.78 | 8.08 | 8.28 |

**Table 3.4.** The comparison of the quality of haplotyping of Linearly Reduced PHASE (LRP) and PHASE (P) vs the original haplotypes (O). Here the difference in haplotype data sets, Hapset1/Hapset2 is the arithmetic mean of numbers of false-positive and false-negative haplotypes over the number of haplotypes Hapset2 times 100%. Each value is averaged over feasible graphs among 25 datasets.

| Datasets | | Average Running Time (seconds) | | | |
|---|---|---|---|---|---|
| | | HAPLOTYPER | Linearly Reduced HAPLOTYPER | | |
| sites | pop | Total | Total | E and D | RH |
| 40 | 20 | 0.0943 | 0.0291 | 0.0019 | 0.0272 |
| 40 | 40 | 0.1578 | 0.0801 | 0.0072 | 0.0729 |
| 80 | 40 | 0.3516 | 0.1335 | 0.0242 | 0.1093 |
| 80 | 80 | 0.6382 | 0.2580 | 0.0415 | 0.2165 |
| 100 | 50 | 0.6631 | 0.1747 | 0.0562 | 0.1185 |
| 100 | 100 | 1.2606 | 0.6035 | 0.0646 | 0.5389 |

**Table 3.5.** The comparison of the running times of HAPLOTYPER and Linearly Reduced HAPLOTYPER. Each value is averaged over 25 datasets.

| Datasets | | Recombination Rate | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | | | 4 | | | 16 | | | 40 | | |
| sites | pop | LRH/H | LRH/O | H/O | LRH/H | LRH/O | H/O | LRH/H | LRH/O | H/O | LRH/H | LRH/O | H/O |
| 40 | 20 | 9.37 | 10.92 | 8.62 | 12.19 | 16.82 | 11.09 | 13.50 | 20.29 | 15.25 | 22.73 | 23.75 | 17.25 |
| 40 | 40 | 5.49 | 7.17 | 3.93 | 7.58 | 8.06 | 5.01 | 6.49 | 6.73 | 7.53 | 11.53 | 13.67 | 10.15 |
| 80 | 40 | 7.46 | 9.38 | 5.23 | 12.93 | 14.67 | 4.34 | 11.37 | 15.98 | 14.54 | 14.33 | 23.81 | 18.53 |
| 80 | 80 | 3.39 | 3.71 | 0.05 | 4.87 | 5.39 | 1.78 | 5.36 | 7.40 | 5.00 | 7.05 | 9.25 | 5.84 |
| 100 | 50 | 8.74 | 8.64 | 3.00 | 10.36 | 14.21 | 7.08 | 9.07 | 13.21 | 9.74 | 15.95 | 19.26 | 14.71 |
| 100 | 100 | 7.45 | 8.16 | 0.008 | 3.91 | 0.004 | 0.005 | 4.74 | 6.72 | 3.95 | 10.91 | 12.45 | 9.29 |

**Table 3.6.** The comparison of the quality of haplotyping of Linearly Reduced HAPLOTYPER (LRH) and HAPLOTYPER (H) vs the original haplotypes (O). Here the difference in haplotype data sets, Hapset1/Hapset2 is the arithmetic mean of numbers of false-positive and false-negative haplotypes over the number of haplotypes Hapset2 times 100%. Each value is averaged over feasible graphs among 25 datasets.

| Datasets | | Recombination Rate | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | | | 4 | | |
| sites | pop | LRH/H | LRH/O | H/O | LRH/H | LRH/O | H/O |
| 40 | 20 | 4.12 | 10.01 | 7.05 | 8.82 | 8.82 | 5.88 |
| 40 | 40 | 6.19 | 8.48 | 5.35 | 2.61 | 5.71 | 5.71 |
| 80 | 40 | 2.38 | 11.36 | 9.09 | 4.11 | 11.80 | 11.10 |
| 80 | 80 | 4.16 | 6.69 | 7.29 | 0.76 | 6.25 | 7.03 |
| 100 | 50 | 4.71 | 7.41 | 9.26 | 4.16 | 9.45 | 5.41 |
| 100 | 100 | 3.48 | 10.46 | 9.30 | 6.96 | 9.58 | 7.50 |

**Table 3.7.** The comparison of the quality of haplotyping of Linearly Reduced HAPLOTYPER (LRH) and HAPLOTYPER (H) vs the original haplotypes (O). Here the difference in haplotype data sets, Hapset1/Hapset2 is the arithmetic mean of numbers of false-positive and false-negative haplotypes over the number of haplotypes Hapset2 times 100%. Each value is averaged over feasible graphs among 25 datasets.

| sites | pop | H | LRH | PHASE | LRP |
|---|---|---|---|---|---|
| 43 | 11 | 0.0903 | 0.00846 | 37.5904 | 2.8928 |
| 43 | 22 | 0.1276 | 0.0192 | 59.8368 | 5.5232 |
| 43 | 33 | 0.1712 | 0.0318 | 105.0596 | 8.6668 |
| 21 | 95 | 0.3250 | 0.2310 | 381.6423 | 321.4914 |
| 50 | 48 | 0.5510 | 0.0720 | 617.3057 | 262.6013 |

**Table 3.8.** The comparison of the running times on real data.

| sites | pop | LRH/H | LRH/O | H/O | LRP/P | LRP/O | P/O |
|---|---|---|---|---|---|---|---|
| 43 | 11 | 8.62 | 11.33 | 8.89 | 7.84 | 11.17 | 8.22 |
| 43 | 22 | 1.99 | 7.11 | 6.22 | 1.79 | 7.89 | 6.88 |
| 43 | 33 | 1.48 | 6.01 | 6.89 | 2.23 | 7.33 | 6.22 |
| 21 | 95 | 6.52 | | | 2.53 | | |
| 50 | 48 | 2.54 | | | 0.29 | | |

**Table 3.9.** The comparison of Linearly Reduced HAPLOTYPER (LRH), HAPLOTYPER(H), Linearly Reduced PHASE (LRP), PHASE (P), and original haplotypes (O) on biological data.

## 3.2 Phasing and Missing data recovery in Family Trios

### 3.2.1 Previous Work and Problem Formulation

In disease association study, family trio data are commonly obtained. genotype data represent family trios consisting of the two parents and their child since that allows to recover haplotypes with higher confidence. A simple logical analysis allows to substantially decrease uncertainty of phasing. For example, for two SNPs in a trio with parent genotypes $f = 22$ and $m = 02$, and the child genotype $k = 01$, there is a unique feasible phasing of the parents: $f_1 = 10$, $f_2 = 01$, $m_1 = 01$, $m_2 = 00$ such that the haplotypes $f_2$ and $m_1$ are inherited by the child. In fact, it is not difficult to check that logical ambiguity exists only if all three genotypes have 2's in the same SNP site (see Figure 3.6.

Usually one can disregard mutations, i.e., if there is no evidence of mutations in children, then phasing should explain genotypes assuming no mutations. Formally, the main problem can be formulated as follows.

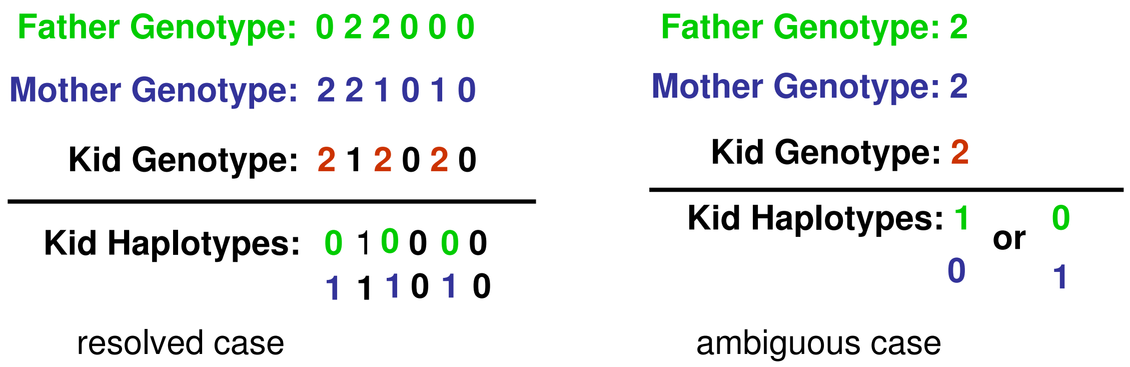


**Figure 3.6.** Resolve child's haplotypes

**Trio Phasing Problem (TPP).** Given a set of genotypes representing family trios, find for each trio a set of six haplotypes maintaining haplotype family trio constraints, i.e., two child haplotypes should be recombination of a pair of (inherited) parental haplotypes – one paternal and one maternal haplotypes.

It is easy to see that the haplotypes of children is much harder to recover than the haplotypes of parents since the family trio data do not provide any direct evidence of recombination between parents' haplotypes inherited by a child. Therefore, we suggest to distinguish the following simpler subproblem of TPP which is concerned only with phasing parents:

**Parental Trio Phasing Problem (PTPP).** Given a set of genotypes representing family trios, find for each trio a quartet of parental haplotypes maintaining genotype family trio constraints, i.e., a pair of parental haplotypes (one paternal and one maternal) should form the given child genotype.

Some researchers [6, 63, 40] introduced several methods for solving above task.

Acherman et al. [6] described the tool Phamily for phasing the trio families based on well-known phasing tool PHASE [87]. It first uses the logical method described above to infer the SNPs in the parental haplotypes. Then children genotypes are discarded while the parental genotypes and known haplotypes are passed to PHASE.

Because the children genotypes are discarded, PHASE no longer can maintain parent-child trio constraints resulting in 8.02% error rate for phasing Daly et al [26] data.

Li et al. [63] developed a computer program consisting of four algorithms for inferring haplotypes from genotypes on pedigree data. These algorithms are designed based on a combinatorial formulation of haplotype inference, namely the minimum-recombinant haplotype configuration (MRHC) problem, and are effective for different types of data. One of the algorithms, called block-extension, is an efficient heuristic algorithm for MRHC that performs very well when the input data requires few recombinants. The other three are exact algorithms for MRHC under various restrictions.

Halperin et al. [40] used the greedy method for phasing and missing data recovery. For each trio the author introduce four partially resolved haplotypes with the coordinates 0, 1 and ?. The values of 0 and 1 correspond to fully resolved SNPs which can be found via logical resolution, while the ?'s corresponds to ambiguous and missing positions. The greedy algorithm iteratively finds the complete haplotype which covers the maximum possible number of partial haplotypes, removes this set of resolved partial haplotypes and continues in that manner. The authors replace each genotype in Daly et al [26] data with a pair of logically partial resolved haplotypes referring to each ambiguous SNP value as a ?. The ?'s constitute 16% of all data. Then extra 10% of data are erased (i.e., replaced with ?'s) and the resulted 26% of ambiguous SNP values are inferred by the greedy algorithm minimizing haplotype variability within blocks. When measured on the additionally erased 10% of data, the error rate for the greedy algorithm is 2.8% [40] which has been independently confirmed in our computational experiments. Unfortunately, the error rate o for the original 16% of ?'s is at least 25% which has been measured by the number of inconsistently phased SNPs. This may lead to a conclusion that the complexity of missing genotype data is considerably higher than the complexity of the successfully genotyped data.

Pasaniuc and Ion Mondoiu [77] propose a highly scalable algorithm based on the entropy minimization principle. They use a local optimization algorithm, which in practice results in genotype phasings with lower entropy. The algorithm achieves a phasing accuracy close to that of best existing methods while being several orders of magnitude faster.

There are no obvious ways to adjust an existing phasing tool to solve PTPP (i.e., phase parents in trios). Let us consider the following strategy (see [6]): (i) use the logical method described above to infer the SNP's in the parental haplotypes, (ii) discard children's genotypes, and (iii) pass to phasing the parental genotypes and known haplotypes. The drawback of this method is that discarding children haplotypes may lead to infeasible phase inferring. Indeed, consider an example with two SNP's (not necessarily adjacent) with the following values: parent genotypes $f = 22$ and $m = 22$, and the child genotype $k = 22$. These two SNP's may violate 4-gamete rule and, therefore, it is possible that paternal haplotypes are resolved as $f_1 = 01$, $f_2 = 10$ while maternal haplotypes are alternatively resolved as $m_1 = 00$ and $m_2 = 11$. Regardless of which paternal and which maternal haplotype would be inherited by a child, it cannot have the required genotype $k = 22$ which implies that parental haplotypes were resolved incorrectly.

We have tried several well-known computational methods [37, 87, 69] for phasing family trio data but all of them give infeasible solutions with an non-negligible inconsistency rate. In other words, current haplotype inference for trios is incorrect not only because it is difficult to resolve the phases but also because it does not simply deliver plausible solutions.

The Table 3.10 gives the results of applying of three phasing methods (GERBIL [37], PHASE [87] and HAPLOTYPER [69]) to the real data of Daly *et al* [26], Gabriel *et al* [32] and simulated data generated by ms [54].

|  | GERBIL |  | PHASE |  | HAPLOTYPER |  |
|---|---|---|---|---|---|---|
| Data | Error % | D % | Error % | D % | Error % | D % |
| Daly *et. al* | 2.1 | 1.2 | 1.1 | 0.5 | 2.2 | 1.2 |
| Gabril *et. al* | 3.1 | 2.5 | 2.2 | 1.3 | 4.3 | 2.7 |
| MS | 7.4 | 6.1 | 9.4 | 6.5 | 8.1 | 5.4 |

**Table 3.10.** The results for three phasing methods on the real data sets [26, 32, 54] and simulated data set. Error% is the percent sites where (best choice of) paternal and maternal haplotypes disagree with the offspring genotype. D % is the Hamming distance between the phased haplotypes and the closest feasible haplotypes.

The ms data were obtained by simulating [54] data – 258 have been generated, each population with 100 individuals and each haplotype with 103 SNPs, then one haplotype has been randomly chosen from each population as a parental haplotype. We obtain family trio haplotypes and genotypes by random matching the generated parental haplotypes.

Each phasing methods has been given as an input the entire genotype population including the genotypes of parents and offspring. The error rate in the column (Error) is measured as follows. For each trio, all four possibilities of inherited parental haplotypes (two choices for paternal and 2 choices for maternal haplotype) are checked and the one which has the minimum number of violations with the offspring genotype is chosen. The percent of SNP sites with violations of parent-offspring constraints out of the total number of SNP sites in offspring genotypes is reported.

### 3.2.2 Pure-Parsimony Trio Phasing

It is easy to find a feasible solution to TPP. However, the number of feasible solutions is exponential and it is necessary to choose a criteria for comparing such solutions. In [63] for haplotyping pedigree data, the objective is to minimize recombinations. That objective is not suitable for TPP since the trios are not full-fledged pedigree data and contain no clues to evidence recombination reconstruction. Thus, following [20, 35], we have decided to pursue parsimonious objective, i.e., minimization of the total number of haplotypes.

**Pure-Parsimony Trio Phasing** (PPTP). Given $3n$ genotypes corresponding to $n$ family trios find minimum number of distinct haplotypes explaining all trios.

The drawback of pure parsimony is that when the number of SNPs becomes large (as well as the number of recombinations), then the quality of pure parsimony phasing is diminishing [35]. Therefore, following the approach in [38], we suggest to partition the genotypes into blocks, i.e., substrings of bounded length, and find solution for the pure parsimony problem for each block separately. Note that in case of family trios we have great advantage over the method of [38] since we do not need to solve the problem of joining blocks. Indeed, for each family trio we can make four haplotype templates (partially resolved by logic means of haplotypes) that imply unique way of gluing together blocks to arrange complete haplotypes for the entire sequence of SNPs.

### 3.2.3 Integer Linear Program for Trio Phasing

As shown in Table 3.10 in the column Error, there is an non-negligible error rate of phasing methods for unrelated individuals (such as PHASE or GERBIL) when applied to family trio data. We suggest to adjust solutions of a phasing method $F$ rather than the phasing method $F$ itself. We suggest to find a feasible solution for TPP which is one of the *closest* to an infeasible solution given by the phasing method $F$. The distance between haplotypes is measured as Hamming distance, i.e., the number of sites where the haplotypes are different. Finding the closest feasible solution can be separately done for each SNP site by checking all possible feasible phase assignment to all six haplotypes and choosing the least different from the original solution given by the method $F$. The average distance to the closest feasible solution for different methods is given in the column $D$ of Table 5.2.

Formally, let *genotype* be a vector with $m$ coordinates each corresponding to an SNP and having one of the following values: 0 (homozygote with major allele), 1 (homozygote with minor allele), 2 (heterozygote), or ? (missing SNP value). Let *haplotype* be a vector with $m$ coordinates where each coordinate is either 0 or 1. We say that two haplotypes *explain* a genotype if

- for any 0 (resp. 1) in the genotype vector, the corresponding coordinates in the both haplotype vectors are 0's (resp. 1's),
- for any 2 in the genotype vector, the corresponding coordinates in the two haplotype vectors are 0 and 1,
- for any ? in the genotype vector, the corresponding coordinates in the haplotypes are unconstrained (can be arbitrary).

We say that four haplotypes $h_1, h_2, h_3, h_4$ *explain* a family trio of genotypes $(f, m, k)$, if $h_1$ and $h_2$ explain the genotype $f$, $h_3$ and $h_4$ explain the genotype $m$, and $h_1$ and $h_3$ explain the genotype $k$.

**Pure-Parsimony Parental Trio Phasing Problem.** (PPTPP). Given $3n$ genotypes corresponding to $n$ family trios find minimum number of distinct haplotypes explaining all trios.

*Integer Linear Programs for Pure-Parsimony* The first two ILP formulations for the PPTPP in this section are inspired by known ILP formulations for phasing from [24] and [15] and the third ILP improves the first two in all three parameters - number of variables, number of constraints and the runtime. We conclude the section with empirical comparison of all three ILP.

The ILP phasing formulation from [24] uses 0-1 variable $x_i$ for each possible haplotype with the minimization objective:

$$\text{Minimize } \sum x_i \tag{3.8}$$

The main drawback of this approach is in exponential number of possible haplotypes which becomes less critical for blocks of limited size. Indeed, if the block size is $b$, then regardless of the number of genotypes, there are at most $2^b$ distinct haplotypes. Although depending on block definition one can find lengthy blocks, e.g. of length 22 [26], $b$ is rarely exceeds 11. Anyway, we can always enforce an appropriate limit on the block size.

The constraints forcing haplotypes to explain given genotypes can be expressed as follows [24]. For any genotype $g$ we introduce a constraint $\sum_{h_i,h_j \text{ explain } g} p_{ij} \geq 1$, where the 0-1 *pair* variable $p_{ij} = 1$ if $g$ can be explained with $h_i$ and $h_j$ in the resulting phasing. That can happen only if the corresponding variables $x_i$ and $x_j$ are set to 1, i.e., $x_i \geq p_{ij}$ and $x_j \geq p_{ij}$.

An obvious adaptation to PPTPP of the above phasing ILP is to have a constraint for each trio $t$

$$\sum_{h_i,h_j,h_k,h_l \text{ explain } t} q_{ijkl} \geq 1 \tag{3.9}$$

where the 0-1 *quartet* variables $q_{i,j,k,l} = 1$ if $t$ can be explained with $h_i, h_j, h_k, h_l$ in the resulting phasing. That can happen only if the corresponding variables $x_i, x_j, x_k, x_l$ are set to 1, i.e.,

$$x_i, x_j, x_k, x_l \geq q_{ijkl} \tag{3.10}$$

The simple ILP (3.8-3.10) has too many variables and constraints and can handle only blocks of size at most 4 for data from [26] (see Table 3.11).

Our second ILP is based on templates of haplotypes. For each trio we introduce four template haplotypes, i.e., haplotypes with the coordinates 0,1,2 and ?. The values of 0 and 1 correspond to fully resolved SNP's which can be found via logical resolution from the previous section, while 2 corresponds to the fact that there is another template with 2 in the same position such that feasible phasing requires these two values be complementary (0 and 1). The ?'s corresponds to free positions. For each 2 in each template we introduce a 0-1-variable $y$ and constraints connecting each pair of complimentary 2's:

$$y + y' = 1 \tag{3.11}$$

For each ? in each template we also introduce a 0-1 variable $z$.

Finally, we need to express dependencies between $x$-variables and $y, z$-variables. Assume that the template $T$ is resolved by the haplotype $h$ if the variables $y_i, i \in I_0$, are set to 0, $y_i, i \in I_1$, are set to 1, $z_j, j \in J_0$, are set to 0, $z_j, j \in J_1$ are set to 1. Then the $x$-variable corresponding to $h$ is constrained as follows

$$\begin{aligned} x \quad \geq \quad & 1 + \sum_{i \in I_1} y_i - |I_1| + \sum_{i \in I_2} (1 - y_i) - |I_2| \\ & + \sum_{j \in J_1} z_j - |J_1| + \sum_{j \in J_2} (1 - z_j) - |J_2| \end{aligned} \tag{3.12}$$

Indeed, if all $y$'s and $z$'s are set as in the haplotype $h$, then all elements of all four sums are 1's and they will be canceled by subtraction of the number of elements in these sums; thus the right hand side is 1. Otherwise right hand side is at most 0 and $x$ is not constrained.

The ILP (3.8)-(3.11)-(3.12) has considerably less variables and constraints than the first ILP but in case of many ?'s there may be too many variables which can slow down ILP solver (see Table 3.11).

Our third ILP takes advantage of the fact that ?'s are really not constrained. Instead of completely resolving templates as in constraint (3.12), we can partially resolve templates, i.e., resolve only 2's. Then several haplotypes can fit partially resolved templates and at least one of the corresponding $x$-variables should be set to 1, i.e., for any $y$-assignment of 2's in each template $T$,

$$\sum_{x \text{ fits all } y'\text{s in } T} x \geq 1 + \sum_{i \in I_1} y_i - |I_1| + \sum_{i \in I_2} (1 - y_i) - |I_2| \tag{3.13}$$

The last constraint is not completely equivalent to (3.12) since now we should guarantee that each template is resolved. This is guaranteed by the following constraint. For each template $T$,

|  |  | Daly Data |  |  |
|---|---|---|---|---|
| Block Size |  | LP1 | LP2 | LP3 |
| 4 | rt (s) | - | 0.103 | 0.0738 |
|  | var | 131783 | 383.73 | 68.76 |
|  | cons | 131896 | 1332.12 | 1428.08 |
| 5 | rt (s) | - | 0.225 | 0.121 |
|  | var | - | 485.28 | 97.5 |
|  | cons | - | 1740.76 | 1663.0 |
| 6 | rt (s) | - | 3.778 | 0.693 |
|  | var | - | 620.12 | 145.82 |
|  | cons | - | 2396.75 | 2036.76 |
| 7 | rt (s) | - | 4688.26 | 2.161 |
|  | var | - | 746.6 | 202.13 |
|  | cons | - | 3193.6 | 2503.4 |
| 8 | rt (s) | - | - | 3.507 |
|  | var | - | - | 5142.85 |
|  | cons | - | - | 598.769 |

**Table 3.11.** The comparison of the running times, number of variables, number of constraints of three linear programs. Each value is averaged over all blocks. All phasing block sizes are uniform.

$$\sum_{x \text{ fits template } T} x \geq 1 \tag{3.14}$$

The ILP (3.8)-(3.11)-(3.13)-(3.14) can resolve longer blocks (see Table 3.11).

### 3.2.4 Greedy Method for Trio Phasing

We apply the greedy algorithm from Halperin [39] for trio phasing. For each trio we introduce four partial haplotypes with the coordinates 0, 1 and ?. The values of 0 and 1 correspond to fully resolved SNPs which can be found via logical resolution, while the ?'s corresponds to ambiguous and missing positions. The greedy algorithm iteratively finds the complete haplotype which covers the maximum possible number of partial haplotypes, removes this set of resolved partial haplotypes and continues in that manner. The drawback of this greed method is to introduce error to trio constraint, even for phasing with error O(m) to the maximum concentration. In the future, we will try to modify the greedy algorithm to overcome its shortcoming.

### 3.2.5 Experimental Results

In this section we compare our greedy and ILP based methods with previously introduced phasing methods such as PHASE[87], PHAMILY[6] and HAPLOTYPER[69] applied to phasing and missing recovery on family trio data. We first describe the test data sets then give experimental results of five methods for phasing and then for missing data recovery of family trio data.

**Data sets.** Our algorithms are evaluated on real and simulated data. The data set collected by Daly *et al.* [26] is derived from the 616 kilobase region of human Chromosome $5q31$ that may contain a genetic variant responsible for Crohn disease by genotyping 103 SNPs for 129 trios. The missing data in this genotype data is about 16%. The data set was partitioned [26] into eleven blocks with only a few common haplotypes inside.

The another real data set is collect by Gabriel *et al.* [32]. This data consists of genotypes of SNPs from 62 region. We use population D which contains of 30 trois from Yoruba. This data set contains about 10% missing data.

The simulated data is generated using ms [54], a well-known haplotype generator based on the coalescent model of SNP sequence evolution. The ms generator emits a haplotype population for the given number of haplotypes, number of SNPs, and the recombination rate. We have simulated Daly *et al.* [26] data by generating 258 populations, each population with 100 individuals and each haplotype with 103 SNPs, then randomly choosing one haplotype from each population. We only simulate parents's haplotypes, then we obtain family trio haplotypes and genotypes by random matching the parental haplotypes.

**Validation of Algorithms.**It is clear how to validate a phasing method on simulated data since the underlying haplotypes are known. The validation on real data is usually performed on the trio data. E.g., a phasing method is applied to parents (respectively, to children) genotypes and the resulted haplotypes are validated on children's (respectively, on parents') genotypes. Unfortunately, in our case, one can not apply such validation since a trio phasing method may rely on both children and parents' genotypes. Therefore, we suggest to validate trio phasing by erasing randomly chosen SNP values and recording the errors in the erased SNP sites.

There are two types of error: true error and logic error. The true error is measured by the Hamming distance from the method's solution to the closest feasible phasing. The logic error is measures by the difference between the method's solution and logic solution. A logical solution allows to substantially decrease uncertainty of phasing. For example, for two SNPs in a trio with parent genotypes $f = 22$ and $m = 02$, and the child genotype $k = 01$, there is a unique feasible phasing of the parents: $f_1 = 10$, $f_2 = 01$, $m_1 = 01$, $m_2 = 00$ such that the haplotypes $f_2$ and $m_1$ are inherited by the child. In fact, it is not difficult to check that logical ambiguity exists only if all three genotypes have 2's in the same SNP site.

we report the percent of SNP values that should be inverted out of the total number of SNP values that should be inferred (i.e., number of 2 plus number of unknown values) based on the logical template. For children (C), we report the percent of SNP which should be inverted with respect to the total number of SNPs. The total number of errors (T) is the percent of SNP's that should be inverted in order to obtain a feasible phasing solution.

In Table 3.12 we compare the logical error for erased data among 5 phasing methods. In the table, each row corresponds to an instance of real data (Daly *et al.* and Gabriel *et al.*) and simulated data (ms) and the column (E) shows the percent of erased data (0% - no data erased, 1%-10% - percent of SNP values erased).

In Table 3.13, we also report true error for phasing simulated genotype data which is the Hamming distance between inferred and actual simulated underlying haplotypes for children (C), for parents (P) and the total error (T).

| | | ILP | | | Greedy | | | Phamily | | | PHASE | | | HAPLOTYPER | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | E | C | P | T | C | P | T | C | P | T | C | P | T | C | P | T |
| Daly *et al.* | 0 | 0.0 | 0.0 | 0.0 | 4.9 | 16.2 | 3.8 | 1.3 | 0.0 | 0.7 | 1.1 | 0.0 | 0.6 | 2.2 | 0.0 | 1.2 |
| | 1 | 0.2 | 0.5 | 0.2 | 4.8 | 16.8 | 3.8 | 1.2 | 1.4 | 0.7 | 1.3 | 0.2 | 0.7 | 2.1 | 1.0 | 1.6 |
| | 2 | 0.3 | 0.7 | 0.4 | 5.0 | 16.9 | 4.0 | 1.3 | 1.8 | 0.9 | 1.3 | 0.5 | 0.8 | 2.2 | 2.3 | 1.7 |
| | 5 | 0.8 | 2.6 | 1.2 | 5.3 | 17.1 | 4.0 | 1.3 | 1.0 | 1.0 | 1.6 | 0.9 | 1.0 | 2.3 | 7.0 | 2.9 |
| | 10 | 1.8 | 6.7 | 3.0 | 5.9 | 17.2 | 4.7 | 1.5 | 2.2 | 1.3 | 1.5 | 1.9 | 1.2 | 2.6 | 9.8 | 4.1 |
| Gabriel *et al.* | 0 | 0.0 | 0.0 | 0.0 | 2.9 | 11.5 | 2.2 | 3.0 | 0.0 | 2.0 | 2.2 | 0.0 | 1.3 | 4.4 | 0.0 | 2.7 |
| | 1 | 0.2 | 0.6 | 0.2 | 2.9 | 12.1 | 2.3 | 3.1 | 0.2 | 2.0 | 2.8 | 0.2 | 1.7 | 4.6 | 1.7 | 1.5 |
| | 2 | 0.3 | 1.2 | 0.5 | 3.2 | 12.2 | 2.4 | 3.3 | 0.4 | 2.1 | 2.9 | 0.6 | 1.8 | 4.9 | 3.1 | 1.6 |
| | 5 | 0.8 | 3.4 | 1.1 | 3.4 | 12.2 | 2.9 | 3.4 | 1.3 | 2.5 | 3.0 | 1.4 | 1.6 | 5.4 | 6.3 | 2.1 |
| | 10 | 1.5 | 6.2 | 1.5 | 4.3 | 12.4 | 3.7 | 3.9 | 2.4 | 2.5 | 3.3 | 3.1 | 2.1 | 6.1 | 15.7 | 6.3 |
| ms | 0 | 0.0 | 0.0 | 0.0 | 2.6 | 13.2 | 1.9 | 9.4 | 0.0 | 4.7 | 5.6 | 0.0 | 6.5 | 8.1 | 0.0 | 5.4 |
| | 1 | 0.3 | 1.0 | 0.4 | 2.9 | 13.5 | 1.9 | 10.1 | 0.8 | 4.3 | 5.8 | 1.2 | 5.4 | 8.4 | 2.2 | 5.6 |
| | 2 | 0.5 | 1.9 | 0.7 | 3.1 | 13.7 | 2.1 | 10.4 | 1.8 | 7.8 | 5.9 | 2.3 | 5.5 | 8.9 | 4.3 | 6.0 |
| | 5 | 1.3 | 3.8 | 1.9 | 4.3 | 13.9 | 3.1 | 10.6 | 3.8 | 7.6 | 6.1 | 4.7 | 5.9 | 9.2 | 10.2 | 7.0 |
| | 10 | 2.5 | 7.7 | 3.6 | 5.3 | 14.0 | 4.4 | 11.9 | 9.5 | 9.2 | 6.9 | 10.5 | 6.0 | 11.5 | 17.1 | 8.0 |

**Table 3.12.** The results for five phasing methods on the real data sets of Daly *et al.*[26], Gabrile *et al.* [32] and on simulated data. The second column corresponds to the ratio of erased data. The C corresponds to the logical error of child. The P corresponds to the logical error of parents. The T corresponds to the total logical error.

| | | ILP | | | Greedy | | | Phamily | | | PHASE | | | HAPLOTYPER | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | E | C | P | T | C | P | T | C | P | T | C | P | T | C | P | T |
| ms | 0 | 1.2 | 1.3 | 1.3 | 1.4 | 1.4 | 1.4 | 2.1 | 2.2 | 2.2 | 3.3 | 3.2 | 3.2 | 2.9 | 2.7 | 2.8 |
| | 1 | 1.3 | 1.3 | 1.3 | 1.3 | 1.4 | 1.4 | 4.5 | 4.0 | 4.3 | 3.2 | 3.3 | 3.2 | 3.0 | 3.2 | 3.1 |
| | 2 | 1.5 | 1.6 | 1.6 | 1.6 | 1.6 | 1.6 | 4.4 | 4.3 | 4.4 | 3.4 | 3.3 | 3.4 | 3.2 | 3.3 | 3.3 |
| | 5 | 2.2 | 2.5 | 2.4 | 2.1 | 2.3 | 2.2 | 4.3 | 4.2 | 4.3 | 3.6 | 3.5 | 3.5 | 3.4 | 3.7 | 3.6 |
| | 10 | 3.0 | 3.7 | 3.5 | 3.3 | 3.3 | 3.3 | 5.2 | 5.2 | 5.2 | 3.1 | 3.0 | 3.0 | 3.9 | 4.2 | 4.1 |

**Table 3.13.** The results for five phasing methods on the simulated data sets. The column E represents the percent of erased data. The C corresponds to the true error of child. The P corresponds to the true error of parents. The T corresponds to the true total error.

Table 3.14 compares five methods (ILP, Greedy, Phamily, PHASE and HAPLOTYPER) on trio missing data recovery on the real data sets (Daly [26] and Gabriel [32]) and simulated data. We erase random data in trio genotypes with certain amount$(1\%, 2\%, 5\%$ and $10\%)$ of the entire data. Instead, We report the error as the number of incorrectly recovered erased positions of the genotypes on child (C*), parents (P*) and trios (T*) divided the total number of erased positions in parent genotypes in percentage. We count only half error if the compared paired SNP is 2 and 0 (or 1).

The result shows that the simple greedy algorithm is very stable in missing data recovery while the ILP method is superior in trio data phasing.

| | | ILP | | | Greedy | | | Phamily | | | PHASE | | | HAPLOTYPER | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | E | C* | P* | T* | C* | P* | T* | C* | P* | T* | C* | P* | T* | C* | P* | T* |
| Daly et al. | 1 | 2.3 | 7.8 | 5.7 | 3.9 | 6.0 | 5.2 | 0.3 | 2.3 | 1.5 | 0.3 | 3.1 | 2.0 | 1.9 | 26.1 | 16.7 |
| | 2 | 3.1 | 8.6 | 6.5 | 4.0 | 6.0 | 5.2 | 0.2 | 4.7 | 3.0 | 0.2 | 3.7 | 2.4 | 1.7 | 24.5 | 15.9 |
| | 5 | 3.9 | 9.9 | 7.8 | 4.5 | 4.8 | 4.7 | 0.2 | 3.6 | 2.5 | 0.1 | 3.4 | 2.3 | 1.3 | 20.5 | 13.9 |
| | 10 | 5.7 | 13.5 | 10.8 | 4.6 | 5.8 | 5.4 | 0.6 | 4.4 | 3.1 | 0.5 | 4.0 | 2.8 | 1.5 | 21.8 | 14.8 |
| Gabriel et al. | 1 | 7.7 | 8.0 | 7.9 | 5.6 | 6.4 | 6.1 | 0 | 2.5 | 1.6 | 0.4 | 3.1 | 2.1 | 1.6 | 21.8 | 14.5 |
| | 2 | 7.1 | 8.6 | 8.1 | 4.9 | 5.7 | 5.5 | 0 | 2.8 | 1.9 | 0.5 | 3.1 | 2.2 | 1.0 | 20.7 | 14.1 |
| | 5 | 7.9 | 8.7 | 8.4 | 5.6 | 5.8 | 5.7 | 0 | 2.3 | 1.5 | 0.1 | 3.3 | 2.2 | 2.5 | 20.7 | 14.6 |
| | 10 | 7.4 | 9.5 | 8.8 | 6.1 | 6.6 | 6.5 | 0.1 | 2.1 | 1.5 | 0.3 | 3.1 | 2.1 | 2.3 | 25.1 | 17.5 |
| ms | 1 | 10.9 | 13.3 | 12.4 | 11.5 | 9.2 | 10.1 | 1.0 | 16.0 | 10.2 | 0.7 | 15.2 | 9.6 | 4.3 | 26.4 | 17.9 |
| | 2 | 11.4 | 12.3 | 11.9 | 11.2 | 8.6 | 9.6 | 1.7 | 15.3 | 10.3 | 0.3 | 15.6 | 10.0 | 4.6 | 20.6 | 14.7 |
| | 5 | 13.1 | 12.1 | 12.4 | 12.3 | 7.8 | 9.3 | 0.9 | 14.8 | 10.0 | 0.7 | 14.9 | 10.0 | 3.6 | 23.1 | 16.4 |
| | 10 | 12.0 | 12.4 | 12.3 | 11.6 | 8.9 | 9.8 | 2.3 | 14.4 | 10.3 | 0.7 | 13.9 | 9.3 | 3.4 | 21.9 | 15.5 |

**Table 3.14.** The results for missing data recovery on the real and simulated data sets with five methods. The second column corresponds to the ratio of erased data. The C* corresponds to the error of child. The P* corresponds to the error of parents. The T* corresponds to the total error.

# CHAPTER 4
# INFORMATIVE SNP SELECTION

The search for the association between complex diseases and single nucleotide polymorphisms (SNPs) has been recently received great attention. For these studies, it is essential to use a small subset of informative SNPs, named *tags*, accurately representing the rest of the SNPs. Firstly, informative SNPs can be used for selective SNP typing and computationally inferring all non-typed SNPs thus achieving considerable budget savings. Secondly, informative SNPs can be used for compaction of SNP data. Indeed, recent successes in high throughput genotyping technologies (e.g., Affimetrix Map Arrays) drastically increase the length of available SNP sequences and they should be compacted to be feasible for fine genotype analysis. This chapter proposes stat-of-the-art informative SNP seleciton tools for applying to disease association study.

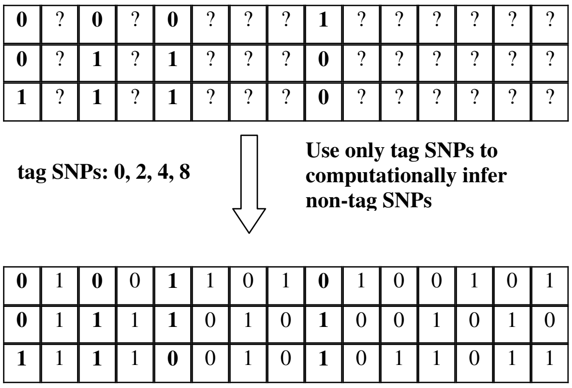
In Section 4.1, we describe previous work on informative SNP selection and formulate the problem. In Section 4.2, we propose linear algebraic methods for solving the problem. In Section 4.3, we show how to separate the tag selection from SNP prediction, formulate the corresponding optimization problem, and describe the general approach and two heuristics for tag selection based on prediction. In Section 4.4, we proposes a new SNP prediction method based on multiple linear regression (MLR) analysis in sigma-restricted coding. In Section 4.5, we proposes a new SNP prediction using a robust tool for classification – Support Vector Machine (SVM). In Section 4.6, we use MLR tagging [46] to reduce set of SNPs. we propose to apply a novel combinatorial method for finding disease-associated multi-SNP combinations.

## 4.1 Previous Work

Informative SNP selection (Tagging) methods have been previously explored in statistical and pattern recognition community as well as optimization community. In statistics, tags are required to *statistically cover* individual (non-tagged) SNPs or haplotypes (sets of SNPs), where the quality of statistical covering is usually measured by correlation, e.g., find minimum number of tags such that for any non-tag SNP there exists a highly correlated (squared correlation $R^2 > .8$) tag SNP [16, 23]. In the optimization community, the number of tags is usually minimized subject to upper bounds on *prediction error* measured as how non-tag SNPs can be predicted from the tag SNPs.

The generic informative SNP selection problem can be formulated as follows (see Figure 4.1:

Given a sample $S$ of a population $P$ of *individuals* (either haplotypes or genotypes) on $m$ SNPs, find positions of $k$ ($k < m$) tag SNPs such that one can predict (or statistically cover) an entire *individual* (haplotype or genotype) from its restriction onto the $k$ tag SNPs.



| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | ? | **0** | ? | **0** | ? | ? | ? | **1** | ? | ? | ? | ? | ? | ? |
| **0** | ? | **1** | ? | **1** | ? | ? | ? | **0** | ? | ? | ? | ? | ? | ? |
| **1** | ? | **1** | ? | **1** | ? | ? | ? | **0** | ? | ? | ? | ? | ? | ? |

**tag SNPs: 0, 2, 4, 8**

**Use only tag SNPs to computationally infer non-tag SNPs**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | **0** | 0 | **1** | 1 | 0 | 1 | **0** | 1 | 0 | 0 | 1 | 0 | 1 |
| **0** | 1 | **1** | 1 | **1** | 0 | 1 | 0 | **1** | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 1 | **1** | 1 | **0** | 0 | 1 | 0 | **1** | 0 | 1 | 1 | 0 | 1 | 1 |

**Figure 4.1.** Problem formulation of Informative SNP Selection

Previous research on tag SNP selection has explored both lossless and lossy methods. Lossless methods select a set of tag SNPs that capture 100% of the haplotypic variation in the sample population. Lossy methods typically select fewer tags than lossless methods, but with some tolerated amount of information loss.

Aviitzhak et al. [8] presented a method for selecting tags which can be used in both a lossless and a lossy manner. The central idea behind both their lossless and lossy methods is to eliminate tags that contribute the least to the Shannon entropy for the haplotype set. First, identical columns and complimentary columns are eliminated, then they eliminate columns that do not reduce the number of unique rows. They note that selecting a maximal linearly independent set of column vectors would miss opportunities to eliminate complimentary SNPs and illustrate that by the 2-by-2 identity matrix[1]. Their lossless method reduces by 25% and 36% the number of SNPs describing the haplotype diversity within an African-American and Caucasian population, respectively.

Zhang et al. [98] introduced a block-based, dynamic programming algorithm for haplotype inference that is capable of reconstructing 90% of the original data using only 35% of SNPs as tags. They used the partition-ligation expectation maximization algorithm [81] for haplotype inference, and as a result, provided a method of performing association studies directly on genotype data.

Sebastiani et al. [83] described a lossless method called BEST (Best Enumeration of SNP Tags) for identifying a minimal set of tag SNPs from haplotype data. BEST selects tags by determining if a candidate tag is a boolean function of SNPs already chosen as tags. The BEST method selected 14% of SNPs as tags from an African-American population and 10% from an European-American population by considering individual genes each ranging from 5 to 229 SNPs in length. However, its effectiveness on a genome-wide scale is still unproven. According to their method, 95% of tags selected from the European-American population were also selected from the African-

[1] In Section 4.2 we show how to adjust linear reduction to avoid such example.

American population, which provides evidence for the a genetic bottleneck event that occurred long ago as hominids migrated out of Africa to settle Europe and Asia.

Halldorson et al. [42] defined the *informativeness* measure of how well a set of tags describes a haplotype sample. Both the informativeness measure, as well as their tag SNP selection method consider a graph whose vertices are SNPs; an edge is placed between to SNPs if one SNP can be used to reliably predict the other. Their method seeks the set of SNPs that maximizes the informativeness measure on the haplotype data. The method can achieve prediction rates of 90% based on only 20% of SNPs. Halldorsson's method differs from the others in that it is a *block-free* method. Block-based methods are restricted to identifying tags only within local contiguous sequences of SNPs where the haplotype diversity is low. Block-free methods have the capability to identify tags across an entire genome. Like Halldorsson's method, the linear reduction method we propose is a block-free method.

Our tagging problem formulations and above approaches do not take into account haplotype frequency when selecting a tag SNPs. For a discussion of how haplotype frequency affects tag SNP selection, see [23, 36, 88].

## 4.2 Linear Algebraic Method

### 4.2.1 Linear Algebraic Tagging

Following Section 3.2.1, if one column-site can be restored from several other columns, then it can be dropped without loss of information. Therefore, we consider restoration of one column-site using a linear combination of other column-sites. Our tagging method is based keeping only linearly independent SNPs as tags. Theorems 3 and 4 show that tagging of haplotype population consisting of recombinations of a limited number of haplotypes can be efficiently reduced to tagging of a small number of linearly independent population representatives.

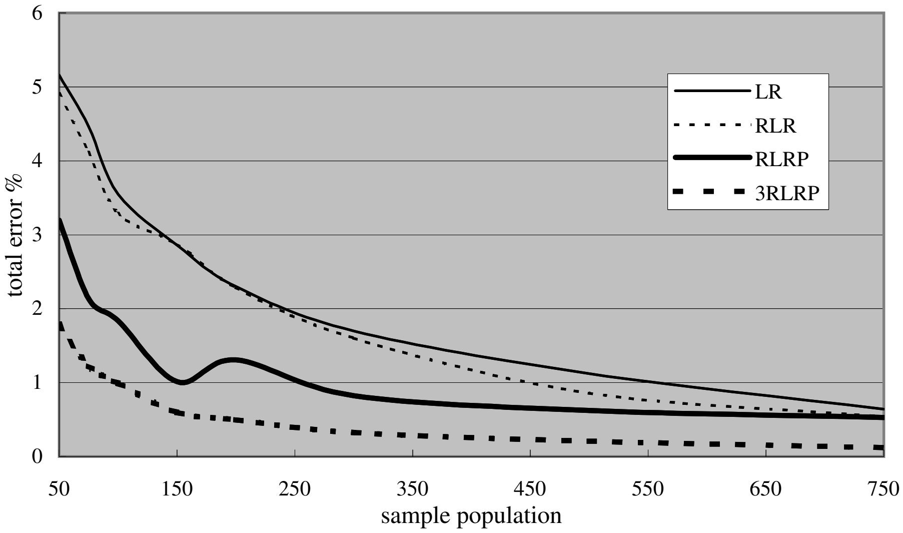


**Figure 4.2.** Simulated data with 25000 sites and haplotype population 1000. The total number of errors in % to the total number of SNPs depending on the size of the sample population for the three algorithms LR, RLR, RLRP and 3RLRP.

From engineering point of view, a deep biostatistical sense of each particular SNP is secondary to the savings in number of tag SNPs. Anyway, the input format of the population $P$ should be decided, and we assume that the population is given as a set of already sequenced haplotypes since any other knowledge about $P$ is inferred and, therefore, arguable. Then the standard experimental way of checking any solution would be picking a random subset $H$ of a set $P$, extracting tag SNP sites and finding reconstruction function based on $H$ and, finally, checking average accuracy rate of prediction over all haplotypes in $P \setminus H$. In order to get a more trustful results, the reported results should be averaged over multiple random choices of $H$. Naturally, the larger the set $H$, the more accuracy can be achieved.

Our sample based linear algebraic tagging consists of the following steps:

From the population sample $H$ extract $r = rank(H)$ of sites $T(H) = \{t_1, \ldots, t_r\}$ forming a basis of columns-sites.

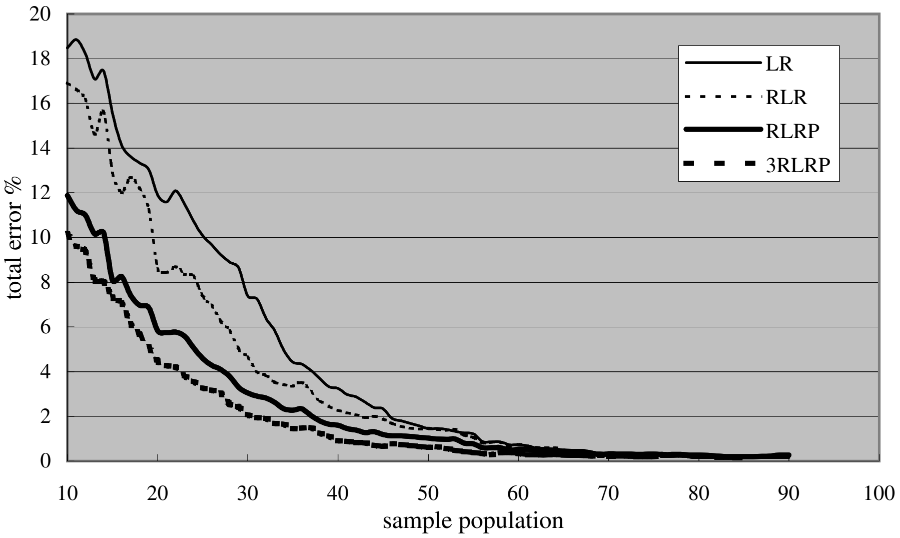


**Figure 4.3.** The dataset of 158 haplotypes with 103 SNPs from [26]. The total number of errors in % to the total number of SNPs depending on the size of the sample population for the three algorithms LR, RLR, RLRP and 3RLRP.

For each column-site in $f_j, j = 1, \ldots, m$ in $H$ find a unique representation $f_j = \sum_{i=1}^{r} \alpha_{i,j} h_{t_i}$

Output the set of tag SNPs $T(H)$ and the reconstruction function $f = (f_1, \ldots, f_m)$.

The suggested linear algebraic method can be implemented very efficiently. Indeed, using $O(n^2m)$ Gaussian elimination, we can transform the $n \times m$ matrix $H$ into the reduced echelon format $H'$ which will have exactly $r$ non-zero rows. The $r$ tag SNPs formed by linearly independent column-sites corresponding to non-zero rows can be easily found from $H'$. Let $F$ be the the matrix $H'$ in which zero rows are dropped, so $F$ is an $r \times m$ matrix. Then for any haplotype $h$ with the tag SNP values $h_r$, the predicted reconstruction $\bar{h} = f(h_r)$ equals

$$\bar{h} = h_r F \tag{4.1}$$

The Gaussian elimination is greedily chooses first linearly independent tag SNPs. Intuitively, the haplotype information is spread all over the haplotype length. There-

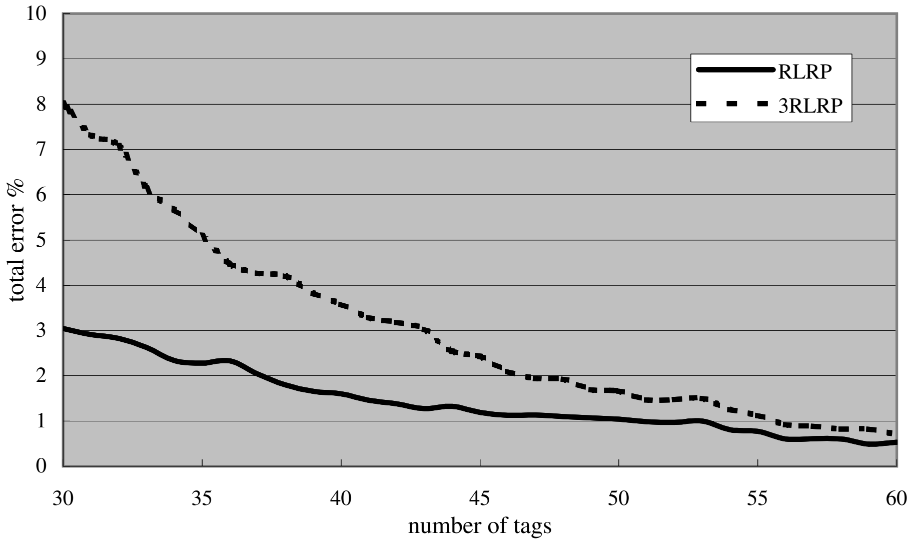


**Figure 4.4.** The dataset of 158 haplotypes with 103 SNPs from [26]. The total number of errors in % to the total number of SNPs depending on the number of the tags for algorithms RLRP and 3RLRP.

fore, we compare two linear reduction (LR) implementations – (i) LR, where the SNPs are taken in the order in which they are given in $H$ and (ii) Randomized LR (RLR), where the SNPs are randomly permuted.

While there is no obvious way to fix falsely predicted SNPs, the unresolved SNPs (i.e., SNP values predicted to be neither $-1$ nor 1) are assigned $-1$ if the predicted values are negative and 1 otherwise. We report the results for the Randomized LR with Postprocessing (RLRP) which is RLR where unresolved SNPs are recovered by the method specified above.

Finally, we report results of the following *3RLRP* method. The RLRP algorithm is run 3 times, three bases are extracted based on the sample and three different predictions of each SNP are obtained. Then out of these three predictions we choose the one in majority. Although 3RLRP uses the same size sample as RLRP, it is based on 3 times more tag SNPs.

For generating the simulated data for sample based tagging, we have used the haplotype generator ms [54]. This generator is a well-known standard based on the

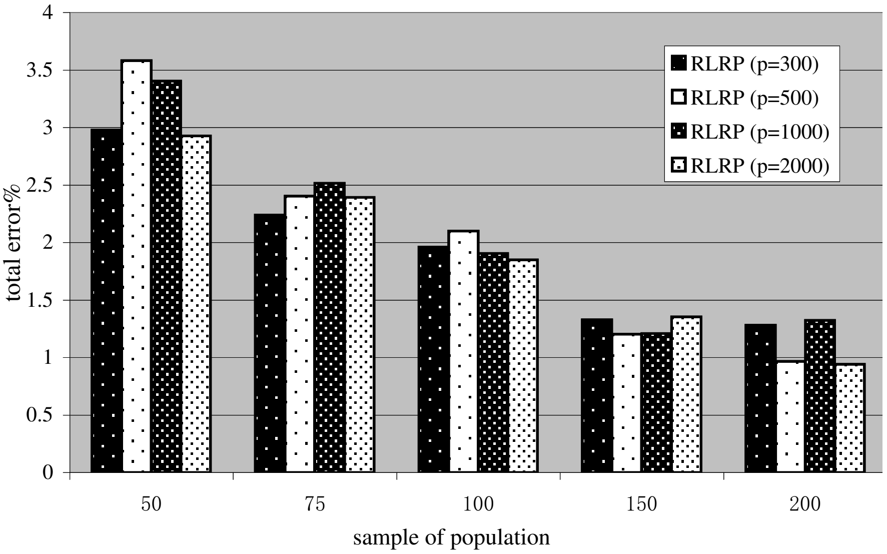


**Figure 4.5.** Simulated data with 25000 sites and different sizes of haplotype population. The total number of errors in % to the total number of SNPs depending on the size of the sample population for the different population sizes ($p = 300, 500, 1000, 2000$).

coalescent model of SNP sequence evolution. The ms generator has capability to generate a given number of haplotypes with the prescribed number of sites and recombination rate. In our tests, we have generated different size haplotype population (300, 500, 1000 and 2000) with 25000 sites based on the recombination rate 40. For population size of 1000 (see Figure 4.2), we report an error depending the size of population sample (50 to 500) while the number of tag SNPs is always close to the size of the sample. The second plot (see Figure 4.7) is devoted to experiments with the real data set derived from the 616 kilobase region of human genotypes [26]. The missing data resolving and haplotyping are taken from [30]. We report an error depending on the size of the population sample while the number of tag SNPs is always less than the size of the sample and comes to 60 for 100 haplotypes. The results are averaged over 10 random draws from the set of all haplotypes. In 3RLRP, the number of tags used is almost three times of that in RLRP. The third plot (see Figure 4.4) compares errors of two algorithms RLRP and 3RLRP for the same number of tag SNPs. The last plot (see Figure 4.5) compares the error rate of RLRP for the same sample size while the population grows from 300 to 2000. As one can see, the error rate does not change with the population size.

### 4.2.2 Linear Algebraic Tagging with Prescribed Number of Tags

In previous section, we use linear reduction method for selecting tag SNPs and reconstructing haplotypes. The LRP algorithm selects linearly independent SNPs from the left side of the haplotype matrix as tags. The RLRP randomly permutes the columns of the haplotype matrix before performing the linear algebraic method. It is shown that the RLRP method has a lower error rate than the LRP method under identical conditions. The 'question' which basis will be the better set' is still open. In this section, we attempt to predict the quality of basis from the corresponding RREF.

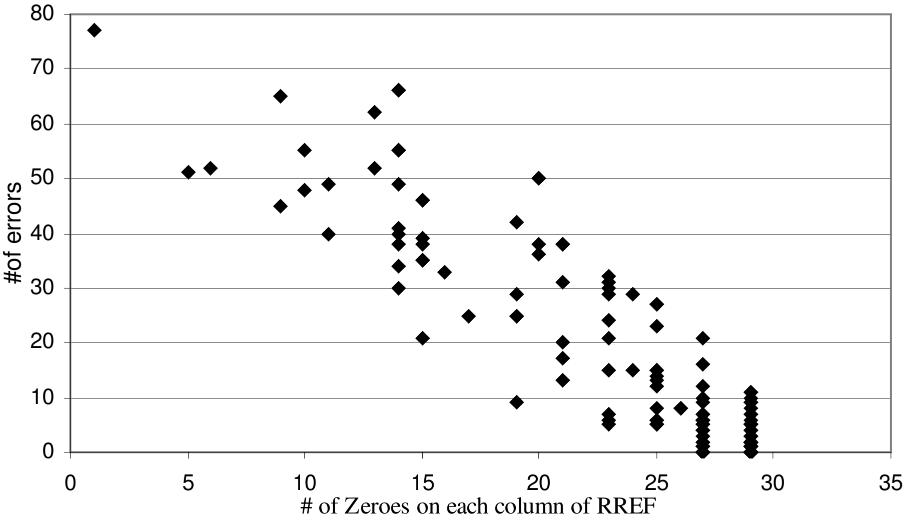


**Figure 4.6.** The x-axis shows the number of zeros in each column of R of the haplotype matrix and the y-axis shows reconstruction error rate for each column in the sample using the RLRP method.

Figure 4.6 plots the number of zeros in each column of the row-reduced echelon form of the haplotype matrix and reconstruction error rate for each column in the sample using the RLRP method. It is easy to see that the number of zeros in RREF is highly correlated to the reconstruction rate. There is an intuitive explanation for this correlation – if in the RREF corresponding to a non-tag SNP contains many zeros, then it can be represented as a linear combination of relatively few tag SNPs. This in term indicates that this linear combination is not result of noise and unlikely to be artefact.

It is computationally infeasible to exhaustively search for the optimal set of linearly independent column-sites which produces the largest number of zeros in the row-reduced echelon form of the haplotype matrix. In stead of exhaustively search, we apply the greed method. We sugggest the following *Separable Linear Algebraic Tagging* (SLT) method. First, the row-reduced echelon form of the haplotype matrix $R = RREF(H)$ is computed using Gauss-Jordan Elimination and the number of zeros in $R$ is calculated. Then, for each column $r_i$ in $R$ containing a pivot, the corresponding column $s_i$ in $H$ is swapped with each column $s_j$ in $H$ that does not contain a pivot and $R = RREF(H)$ is recomputed. If the number of zeros has increased, then we keep the swap, otherwise we swap the columns back to their original positions. The algorithm stops when no swap gives increased number of zeros. the rearranged columns of $H$ gives the largest number of zeros in $R$.

The SLT method is not an exhaustive search since it doesn't test every linearly independent subset of columns of size $r = rank(H)$, and it is possible that when the algorithm stops the optimal subset has not been achieved. In practice, however, the SLT algorithm find the close to optimal solutions.

The haplotype reconstruction algorithm can be separated for the case (i) when the required number of tags $k$ is greater or equal to the rank of the sample $r$ and (2) when the sample rank $r$ is small. We found out that while finding additional tags in case (i) works well, their method does not work as good for the second case. Here, we first describe their voting algorithm for the case (i) and then show how to reduce the case (i) to the case (ii) by linear algebraic means. Finally, based on the reconstruction algorithm for the case (ii), we suggest a new way how to handle missing tags.

**Case (i):** $k \geq r = rank(S)$. We can repeatedly choose $k - r + 1$ different subsets of $r$ linearly independent tags and reconstruct all other SNP using the basic linear algebraic method. This can be done by placing $k$ tag columns to the beginning of the sample matrix $S$, $k - r + 1$ times randomly permuting these $k$ tags and finding RREF of the resulted matrices. Since there are $k - r + 1$ different RREF's, there that many different reconstructions for each haplotype. The aggregation of all these reconstructions is suggested to be done by "voting": each SNP attained the value of $-1$ (respectively, 1) if the majority of $k - r + 1$ reconstructions suggests $-1$ (respectively, 1).

**Case (ii):** $k < r = rank(S)$. According to the basic reconstruction algorithm, the tag vector of the given haplotype/genotype should multiplied by this RREF to obtain reconstruction of the full unknown haplotype/genotype vector. As a result the reconstruction is poor since it does not exploit all information from the sample $S$. Indeed, a single RREF obtained from $S$ and uses only to the first $k$ rows. Here, we suggest just to reuse the algorithm from the case (i) after transposing the matrix $S$. Indeed, for the transposed matrix $S^T$, we get into the same problem as in case (i) – how to choose $k$ columns out of $r = rank(S^T)$ columns. First we need to correctly choose $r - k$ additional columns corresponding to the different sample haplotypes. Here we repeat the procedure used for selecting additional tags – find the columns that have the worst reconstruction by first $k$ haplotypes. These haplotypes will have the maximum number of non-zero values in RREF of $S^T$.

Reconstruction algorithm is also very similar to the case (i). We first find $r$ linear independent rows of $S$ and use RREF of $S$ to predict the given haplotype/genotype. After that the rows of $S$ are permuted $r - k$ times and more $r - k$ different reconstruction of the given haplotype/genotype are obtained. Finally, the aggregation of all these $r - k + 1$ reconstructions is done by the same "voting" as in case (i).

**Handling missing tag SNPs.** If genotyping of tag SNPs of unknown haplotype/genotype contains ?'s, i.e., lacks certain values, then it can be handled in the same way as above. We just simply treat such event as if we would never expected to know these tags. If as a result the number of tags becomes less than the rank, we use the reconstruction algorithm from case (ii), otherwise we use algorithm from case (i).

**The detailed tagging algorithm is as follows:**

---

**Input:** The $s \times m$ sample haplotype matrix $H_s$ with s haplotypes and m SNPs, and an integer number $k$

**Output:** Positions of $k$ tag SNPs $t_1, \ldots, t_k$ and the reconstructing matrix $F$

---

1. For each row $i$ in $H_s$, extract the $i$th row to form the new matrix $H_{s-1,i}$. Use linear reduction on row $i$ to produce a single tag $T_i$ and reconstruction function $F_i$. Predict $H_{s-1}$ from tag $T_i$ and reconstruction function $F_i$.

2. Compute the reconstruction error (total number of differences between original and reconstructed matrix). Delete the row with the most accurate reconstruction from $H_s$.

3. Append the deleted row onto $F$. Perform Gauss-Jordan elimination on $F$ to obtain the reduced-row echelon form $F_r$. If rank($F_r$) remains the same, then delete this row from $F$.

4. (a) If number of rows in $F = k$, output $F$ and tags as linearly independent column in $F_r$

   (b) If number of rows in $F < k$ and $H_s$ is not empty, go to step 1.

```
    (c) If number of rows in $F < k$ and $H_s$ is empty, only repeat step 1. Count number of errors in each column. Sort columns with more errors from left to right. Output the sorted $F$ and $k$ leftmost tags.
```

**The detailed reconstruction algorithm is as follows:**

**Input:** The $p \times m$ haplotype matrix $H_p$, and the $k$ tag sites $t_1, \ldots, t_k$ and the $l \times m$ reconstruction matrix $F$

**Output:** The $s \times m$ predicted haplotype matrix $\bar{h}$

```
1 Perform Gauss-Jordan elimination on $F$ to form rref($F$). Let r = rank(rref($F$)).
2 If $r = k$
    (a) Form the $p \times k$ matrix $H_k$ from $H_p$ with $k$ tag SNP values.
    (b) $\bar{H} = H_k \times rref(F)$
3 Else if $r < k$
  RLRP
    (a) Form tag matrix $T$ defined as $T_{i,j} = F_{i,t_j}$.
    (b) For each non tag SNP $s_q : s_1, \ldots, s_{m-k}$
        (i) Choose the $r$ closest tag SNPs to $s_q$ to form $T_q$. Append $s_q$ to $T_q$.
        (ii) Perform Gauss-Jordan elimination on $T_q$ to form rref($T_q$).
        (iii) Form the $p \times r$ matrix $H_r$ from $H_p$ with $r$ tag SNP values.
        (iv) $\bar{H} = H_r \times rref(T_q)$
    (c) Concatenate all predicted non tag SNPs to form $\bar{H}$ .

VRLRP

    Initialize the $l \times (m-k)$ matrix $V$ to all zeros. For $i$ from 1 to $k-r+1$
        (i) Form the matrix $T_i$ consisting of SNPs $i$ through $i+r$.
        (ii) Form the matrix $F^{'}$ by starting with $T_i$, then appending non tag SNPs
             $s_1, \ldots, s_{m-k}$ on the right side.
        (iii) Perform Gauss-Jordan elimination on $F^{'}$ to form rref($F^{'}$).
        (iv) Form the $p \times r$ matrix $H_r$ from $H_p$ with $r$ tag SNP values.
        (v) $\bar{H} = H_r\times$ rref($F^{'}$). Let $V_i$ be the rightmost $m-k$ columns of $\bar{H}$.
        (vi) Let $V = V + V_i$.
```

### 4.2.3 Experimental Results

We compare the SLT method with the methods described in [43, 45, 42, 98, 36] on the following four data sets:

**Chromosome 5q31.** The Daly et al. [26] data set consists of 516 haplotypes containing 103 SNPs that are derived using trio phasing described in Brinza et al. [13] from 129 family trios on a 616 kilobase region of human chromosome $5q31$ that may contain a genetic variant responsible for Crohn's disease. For each trial, sample data sets of size 10-100 are extracted from the data, and missing data is introduced into the remaining haplotypes at 0%, 5%, and 10% of the total number of SNPs. The SLT method as described in Section 4.2.2 is used to reconstruct the remaining haplotypes. The results are averaged over 10 trials and are reported as the average hamming distance between the reconstructed and real haplotypes as a percentage of the haplotype length (see Figure 4.7). We compare the SLT method with the methods of He et al. [43, 45] on this data set. The linear algebraic method reliably (error rate below 2%) recovers all SNPs based upon a very small portion of tag SNPs (e.g., 32 tags out of total 103 SNPs) while sampling below 8% of the population.

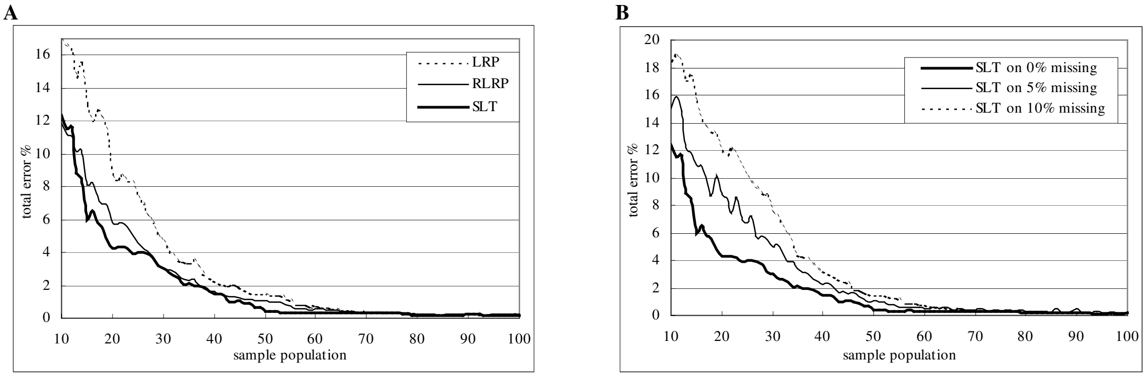


**Figure 4.7.** (A) The total number of errors as a percentage of the total number of SNPs depending on the size of the sample population for the three algorithms LRP, RLRP, and SLT on Chromosome 5q31. (B) The total number of errors as a percentage of total number of SNPs depending on the size of the sample population and the percentage of missing data for the SLT method on Chromosome 5q31.

**LPL & Chromosome 21.** The Clark et al. [21] data set consists of the haplotypes of 71 individuals typed over 88 SNPs in the human lipoprotein lipase (LPL) gene. The Chromosome 21 data set consists the first 1, 000 of 24, 047 SNPs typed over 20 haploid copies of human Chromosome 21 [78]. This subset was found to be highly representative to the entire data set. Both the LPL and Chromosome 21 data sets were used in Halldorsson et al. [42]. We compare the SLT method with the methods of Halldorsson et al. [42] and Zhang et al. [98] on these two data sets by using leave-one-out cross-validation to evaluate the quality of the solution.

The haplotype left out is reconstructed by the SLT method. The Hamming distance between the reconstructed haplotype and the leave-one-out haplotype is recorded; the error rate is the number of errors in the reconstruction as a percentage of the haplotype length. The average error rate in reconstructing all of the haplotypes is used as a measure of the overall accuracy of the tagging method on the data set. The methods of Zhang et al. [98] and Halldorsson et al. [42] impute a SNP based on the tag SNPs in the same block or neighborhood. Therefore, if there is no tag SNP in the block or neighborhood, then these methods do not predict a value for the SNP. The SLT method reconstructs each SNP based on the values of *all* tag SNPs. Figure 4.13 compares the SLT method with the methods of Zhang et al. [98] and Halldorsson et al. [42]. The figure shows that, for the LPL and Chromosome 21 data when 10% of SNPs are used as tags, the SLT method reaches 80% accuracy, while the methods of Halldorsson et al. [42] and Zhang et al. [98] only reach 20% accuracy.

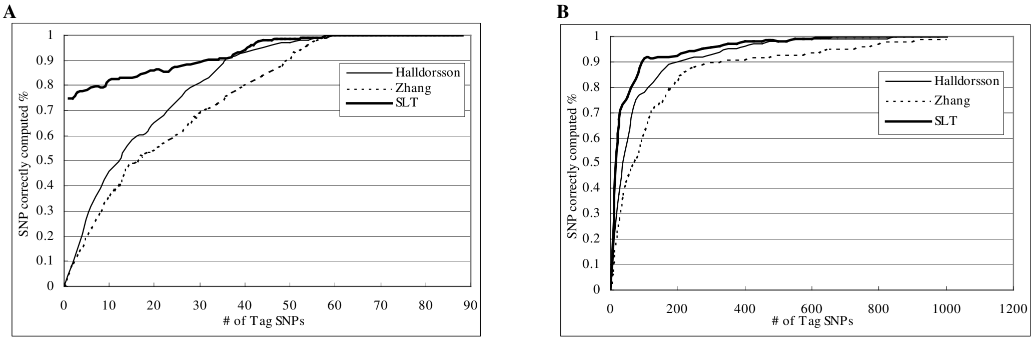


**Figure 4.8.** The x-axis shows the number of tag SNPs, and the y-axis shows the fraction of SNPs correctly imputed in a leave-one-out experiment. (A) Results from SLT, Halldorsson et al. and Zhang et al. for the LPL data set. (B) Results from the SLT method, Halldorsson et al. and Zhang et al. for the Chromosome 21 data set.

**Large simultated data sets.** The Forton et al. [36] data set consists of 32 family trios of European descent genotyped over 122 SNPs across 654 kilobase of the 5q31 cytokine gene cluster. SNPs that violate the Hardy-Weinberg law and SNPs with frequency less than 5% are cleansed from the population. After cleansing, the data set contains 99 SNPs. 95 haplotypes and their frequencies are computed using Phamily and PHASE as described in [87]. To create model populations with which to conduct haplotype reconstruction experiments, first an initial population of $100,000$

individuals is created by selecting random pairs of haplotypes, while ensuring that the haplotype frequencies remained unchanged. From this initial population, 5 smaller populations consisting of 760 unrelated individuals are created by randomly selecting individuals. Five levels of missing data (1%, 2%, 5%, 10% and 20%) are introduced to the population.

The method of Ackerman et al. [6] identifies 22 tag SNPs for the 5q31 cytokine gene cluster. To reconstruct the haplotypes of each population, we assume that the tag values for the genotypes are known. We obtain the tag values of the haplotypes by SNPHAP. In Forton et al. [36], the complete haplotype is extrapolated from the tags by finding the haplotype in the sample with the minimum Hamming distance between its restricted haplotype and the tags. In contrast to Forton et al. [36], we reconstruct the complete haplotype using the linear algebraic method on the same set of tags; the final haplotype is identified as the haplotype in the sample with the minimum Hamming distance to the reconstructed haplotype. The error is recorded as the percentage of incorrect haplotype reconstructions (see Figure 4.9A.). The SLT method is computational more expensive but obtains higher reconstruction accuracy. For example, at the 20% missing level, we obtain 2% higher reconstruction accuracy. Figure 4.9B shows the percentage of error at each haplotype locus over all simulated populations with different levels of missing data.

## 4.3 Tag SNP Selection and SNP Prediction Problems

In this section we show how to separate the tag selection from SNP prediction, formulate the corresponding optimization problem and describe the general approach and two heuristics for tag selection based on prediction.

A *SNP prediction algorithm* $A_k$ accepts as its input the values of $k$ tags $(t_1, \cdots, t_k)$ of an individual $x$ along with the known sample $S$, in which all of the SNP values for

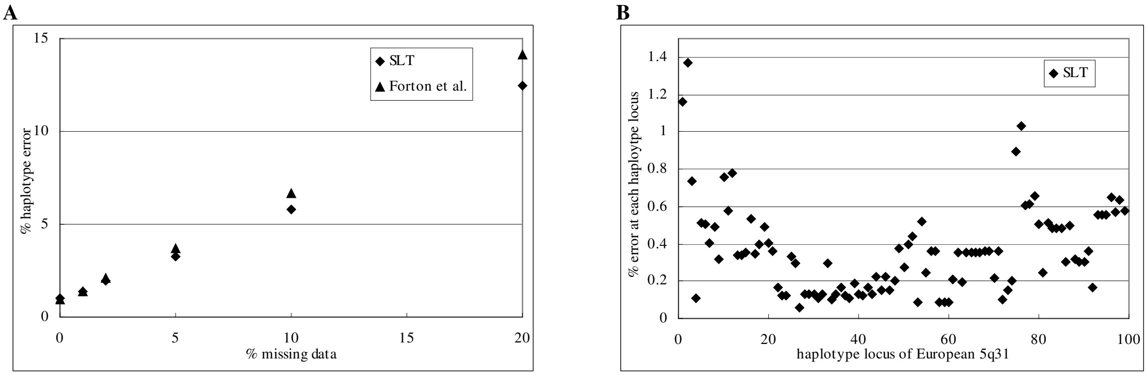


**Figure 4.9.** (A) The x-axis shows the percentage of missing data, and the y-axis shows the percentage of incorrect haplotype reconstructions. Results are from the simulated data sets. (B) The percentage of errors at each haplotype locus over all simulated populations with different levels of missing data of the simulated data sets.

each individual in $S$ is known. The output of $A_k$ is the reconstruction of $x$, that is, $A_k$ predicts the values of each of the non-tag SNPs in $x$.

Assuming self-similarity of data, one can expect that an algorithm predicting with high accuracy a SNP of an unknown individual will also predict with high accuracy SNPs of the sampled individual. Then, we expect that the better prediction algorithm will have fewer errors when predicting SNPs in the sample $S$. This expectation allows to find tags using prediction algorithm as follows: We can check each $k$-tuple of tags and choose the $k$-tuple with the minimal number of errors in predicting the non-tag SNPs in the sampled individuals. Even though the sample elements are completely typed, prediction algorithms can make still errors because the number of SNPs is not sufficient to distinguish any two sampled individuals. Thus, tag SNP selection based on prediction is reduced to the following problem:

**Tag SNP Selection Problem (TSSP).** *Given a prediction algorithm $A_k$ and a sample $S$, find $k$ tags such that the prediction error $e$ of $A_k$ averaged over all SNPs in $S$ (including tags) is minimized.*

Halperin et al.[41] proposed a certain prediction algorithm for which the TSSP problem can be solved exactly in polynomial time using a dynamic programming algorithm STAMPA. In general, this problem is computationally difficult and the runtime of an exact algorithm may become prohibitively slow. Therefore, one can use heuristics for the selection of $k$ tags following Halperin et al.[41] who compare relatively slow STAMPA with a fast random tag selection.

In general, these problems are computationally difficult and the runtime of an exact algorithm may become prohibitively slow. Below we propose two universal heuristics which can be applied to an arbitrary prediction algorithm or statistical covering criteria $A_k$.

The *Stepwise Tagging algorithm* (STA) starts with the best tag $t_0$, i.e., tag that minimizes error when predicting with $A_k$ all other tags. Then STA finds such tag $t_1$ which would be the best extension of $\{t_0\}$ and continue adding best tags until reaching the set of tags of the given size $k$. STA produces *hereditary* set of tags, i.e., the chosen $k$ tags contain the chosen $k-1$ tags. This hereditary property may be useful in case if the set of tags can be extended.

The *Local-Minimization Tag Selection algorithm* (LMT) is more accurately searching for a better set of tags among much larger possibilities. LMT starts with the $k$ tags produced by STA and then iteratively replaces each single tag with the best possible choice while not changing other tags. Such replacements will be continued until no significant improvement in the prediction quality (i.e., by more than given amount of $\epsilon\%$) can be achieved. The runtime of LMT is $O(knmT\epsilon^{-1})$ since the number of iterations cannot exceed $\frac{100}{\epsilon}$.

## 4.4 Multiple Linear Regression SNP Prediction Method

In this section, we propose a new SNP prediction method based on multiple linear regression (MLR) analysis in sigma-restricted coding. When predicting a non-tag SNP, the MLR method accumulates information about all tag SNPs resulting in significantly higher prediction accuracy with the same number of tags than for the previously known tagging methods. We also show that the tag selection strongly depends on how the chosen tags will be used – advantage of one tag set over another can only be considered with respect to a certain prediction method. Two simple universal tag selection methods have been applied: a (faster) stepwise and a (slower) local-minimization tag selection algorithms. An extensive experimental study on various datasets including 10 regions from HapMap shows that the MLR prediction combined with stepwise tag selection uses significantly fewer tags (e.g., up to two times less tags to reach 90% prediction accuracy) than the state-of-art methods of Halperin et al. [41] for genotypes and Halldorsson et al. [42] for haplotypes, respectively. Our stepwise tagging matches the quality of while being faster than STAMPA [41].

### 4.4.1 Introduction to Multiple Linear Regression

The general purpose of multiple linear regression is to learn the relationship between several independent variables and a response variable. The multiple linear regression model is given by

$$\mathbf{y} = \beta_0 + \beta_1\mathbf{x}_1 + \beta_2\mathbf{x_2} + ... + \beta_k\mathbf{x}_k + \epsilon = \mathbf{X}\beta + \epsilon \tag{4.2}$$

where $\mathbf{y}$ is the response variable (represented by a column with n coordinates ($k \leq n-1$)), $\mathbf{x_i}, i = 1, \ldots, k$ are independent variables (columns), $\beta_i, i = 1, \ldots, k$ are regression coefficients, and $\epsilon$ (a column) is the model error. The regression coefficient $\beta_i$ represents the independent contribution of the independent variable $x_i$ to the prediction of $\mathbf{y}$.

### 4.4.2 The MLR SNP Prediction Algorithm

The tags are selected based on the sample population with intention to derive conclusions about the entire population. Statistical analysis may ensure that high prediction quality of non-tag SNPs is not a coincidence. If certain SNP's are highly correlated (i.e. in linkage disequilibrium) in the sample, then we would expect that this correlation will be observed in the entire population. Therefore, it would be highly desirable that the tags contributing to non-tag SNP prediction will correlate with the predicted SNP.

Usually, a genotype is represented by a vector with coordinates 0,1, or 2, where 0 represents the homozygous site with major allele, 1 represents the homozygous site with minor allele, and 2 represents the heterozygous site. Respectively, each haplotype's coordinate is 0 or 1, where 0 represents the major allele and 1 represents the minor allele. The sample population $S$ together with the tag-restricted individual $x$ are represented as a matrix $M$. The matrix $M$ has $n+1$ rows corresponding to $n$ sample individuals and the individual $x$ and $k+1$ columns corresponding to $k$ tag SNPs and a single non-tag SNP $s$. All values in $M$ are known except the value of $s$ in $x$. In case of haplotypes, there are only two possible resolutions of $s$, namely, $s_0$ and $s_1$ with the unknown SNP value equal to 0 or 1, respectively. For genotypes, there are 3 possible resolutions $s_0$, $s_1$, and $s_2$ corresponding to SNP values 0, 1, or 2, respectively. The SNP prediction method should chose correct resolution of $s$.

The proposed MLR SNP prediction method considers all possible resolutions of $s$ together with the set of tag SNPs $T$ as the vectors in $(n+1)$-dimensional Euclidean space. It assumes that the most probable resolution of $s$ should be the "closest" to $T$. The distance between resolution of $s$ and $T$ is measured between $s$ and its projection on the vector space $span(T)$, the span of the set of tag SNPs $T$ (see Figure 4.10).

The MLR method computes $b_i, i = 1, \ldots, k$ estimating unknown *true coefficients* $\beta_i, i = 1, \ldots, k$ minimizing the error $||\epsilon||$ using the least squares method. Geometrically speaking, in the *estimation space* $span(X)$, which is the linear closure of vectors $x_i, i = 1, \ldots, k$, we find the vector $\hat{y} = b_0 + b_1x_1 + b_2x_2 + ... + b_kx_k = Xb$ estimating $y$. The vector $\hat{y}$ minimizing distance (error) $||\epsilon|| = ||\hat{y} - y||$ is the projection of $y$ on $span(X)$ and equals

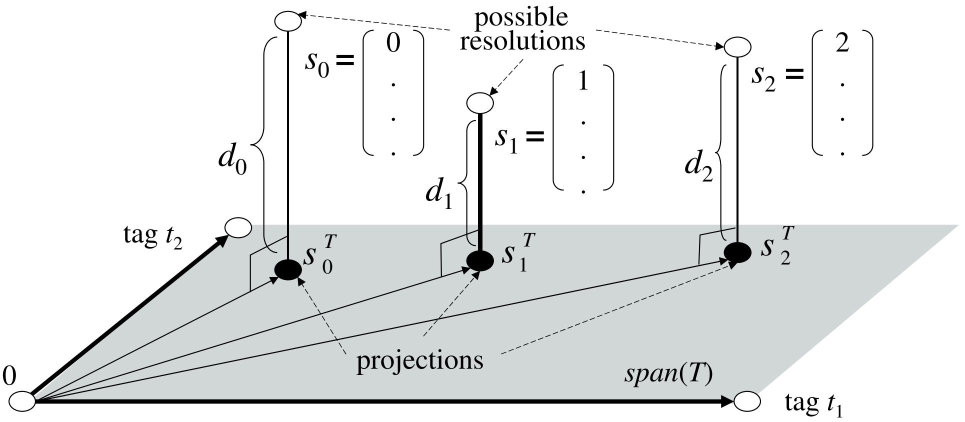


**Figure 4.10.** MLR SNP Prediction Algorithm. Three possible resolutions $s_0$,$s_1$, and $s_2$ of $s$ are projected on the span of tag SNPs (a dark plane). The unknown SNP value is predicted 1 since the distance between $s_1$ and its projection $s_1^T$ is the shorter than for $s_0$ and $s_2$.

$$\hat{y} = X(X^tX)^{-1}X^ty \tag{4.3}$$

Given the values of independent variables $x^* = (x_1^*, \ldots, x_k^*)$, the MLR method can predict (estimate) the corresponding response variable $y^*$ with

$$\hat{y}^* = x^*(X^tX)^{-1}X^ty \tag{4.4}$$

Formally, let $T$ be the $(n+1) \times k$ matrix consisting of $n+1$ rows corresponding to a tag-restricted genotype $x = (x_1^*, \ldots, x_k^*)$ and $n$ sample genotypes $x_i, i = \overline{1,n}$, from $X$, $g_i = \{x_{i,1}, \ldots, x_{i,k}\}$, whose $k$ coordinates correspond to $k$ tag SNPs. The SNP $s$ is represented by a $(n+1)$-column with known values $y_i, i = \overline{1,n}$, for genotypes from $X$ and the unknown value $y^*$ for the genotype $g$ which should be predicted. Let $d = ||\epsilon||$ be the least square distance between $s$ and $T$, i.e., $d = |T \cdot (T^t \cdot T)^{-1} \cdot T^t \cdot s - s|$. Our algorithm finds the value (-1, 0 or 1) for $y*$ and selects one minimizing $d$.

$$T = \begin{bmatrix} x_1^* & \ldots & x_k^* \\ x_{1,1} & \ldots & x_{1,k} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,k} \end{bmatrix} \qquad s = \begin{bmatrix} y^* \\ y_{1,k+1} \\ \vdots \\ y_{n,k+1} \end{bmatrix}$$

### 4.4.3 Running Time of MLR SNP prediction and Tag Selection

Computing of $T^t \cdot T$ is $O(nk^2)$ since T is a $n \times k$ matrix and $T^t$ is a $k \times n$ matrix. For inverting the $k \times k$ matrix $T^t \cdot T$, we need $O(k^3)$ steps. MLR restricts the number of tags $k$ to be less than the number of individual $k \leq n-1$, therefore, the running time for computing $T \cdot (T^t \cdot T)^{-1} \cdot T^t$ is $O(n^2k)$. The matrix of $T \cdot (T^t \cdot T)^{-1} \cdot T^t$ is the same for all these (m-k) non-tag SNPs, thus, the total running time for predicting a complete individual is $O(kn^2 + n^2(m-k)) = O(n^2m)$. We use stepwise tag selection based on multiple linear regression. Therefore, the MLR SNP prediction need knm steps for prediction, thus, the total runtime of MLR-tagging is $O(knmT) = O(knm \times n^2m) = O(kn^3m^2)$.

### 4.4.4 Experimental Results

The following datasets are used to measure the quality of our algorithms. Currently, our algorithms cannot tolerate missing data. Following Halperin et al.[41], we use GERBIL [37] to phase the genotypes and then combine the resulting two haplotypes to recover any missing data.

**Seven ENCODE regions.** Three regions (ENm013, ENr112, ENr113) from 30 CEPH family trios obtained from HapMap ENCODE Project [5]. The number of SNPs genotyped in each region is 361, 412 and 515 respectively. Regions ENr123 and ENm010 from 2 population: 45 singles Han Chinese (HCB) and 44 singles Japanese(JPT). The number of SNPs genotyped in each region is 63 and 105.

**Two gene regions.** Two gene regions STEAP and TRPM8 from 30 CEPH family trios obtained from HapMap [5]. The number of SNPs genotyped in each gene region is 23 and 102 SNPs.

**Chromosome 5q31.** The data set collected by Daly et al. [26] derived from the 616 kilobase region of human Chromosome $5q31$ from 129 family trios.

**LPL & Chromosome 21.** The Clark et al. [21] data set consists of the haplotypes of 71 individuals typed over 88 SNPs in the human lipoprotein lipase (LPL) gene. The Chromosome 21 data set consists the first $1,000$ of $24,047$ SNPs typed over 20 haploid copies of human Chromosome 21 [78].

**Table 4.1.** The quality of SNP prediction from the given number of tags (5% to 15% of the total number of SNPs (in Parentheses). The prediction quality is measured by the prediction accuracy and the average and minimum $R^2$. Total number of SNPs in each dataset is in the parenthesis.

| Tags | Measure | ENm013<br>(360) | ENr112<br>(411) | ENr113<br>(514) | STEAP<br>(22) | TRPM8<br>(101) | 5q31<br>(103) |
|---|---|---|---|---|---|---|---|
| 5% | accuracy | 0.988 | 0.949 | 0.999 | 0.910 | 0.893 | 0.885 |
| | avg $R^2$ | 0.639 | 0.832 | 0.832 | 0.425 | 0.596 | 0.688 |
| | min $R^2$ | 0.486 | 0.433 | 0.630 | 0.010 | 0.001 | 0.096 |
| 10% | accuracy | 0.999 | 0.994 | 1 | 0.969 | 0.947 | 0.935 |
| | avg $R^2$ | 0.959 | 0.972 | 0.972 | 0.638 | 0.736 | 0.794 |
| | min $R^2$ | 0.632 | 0.825 | 0.789 | 0.015 | 0.005 | 0.181 |
| 15% | accuracy | 1 | 1 | 1 | 0.983 | 0.963 | 0.952 |
| | avg $R^2$ | 1 | 1 | 1 | 0.748 | 0.829 | 0.845 |
| | min $R^2$ | 1 | 1 | 1 | 0.017 | 0.009 | 0.248 |

We report the prediction accuracy and the squared correlation $R^2$ between predicted and original non-tag SNP values. The prediction accuracy is measured as the percentage of correctly predicted SNP values on non-tag SNPs. Table 4.6 reports the prediction accuracy and the average and the minimum correlation $R^2$ for all non-tag SNPs.

Alternatively, we apply leave-one-out cross-validation to evaluate the quality of the MLR-Tagging solution for the Genotype Tagging Problem as follows: (1) one by one, each genotype vector is removed from the sample, (2) tag SNPs are selected using only the remaining genotypes, and (3) the "left out" genotype is reconstructed based on its tag SNPs and the values of tag and non-tag SNPs in the remaining genotypes. In Table 4.7, we compare MLR with STAMPA and LR. Note that if one predicts each SNP as 0 (i.e., homozygous with major allele), then the prediction accuracy on

**Table 4.2.** Number of tags used by MLR-tagging, STAMPA and LR to achieve 80% and 90% prediction accuracy in leave-one-out tests.

| Acc. | Algorithm | ENm013<br>(360) | ENr112<br>(411) | ENr113<br>(514) | STEAP<br>(22) | TRPM8<br>(101) | 5q31<br>(103) |
|---|---|---|---|---|---|---|---|
| 80% | MLR | 2 | 6 | 4 | 1 | 1 | 1 |
| | STAMPA | 5 | 9 | 11 | 2 | 3 | 2 |
| | LR | 11 | 17 | 35 | 4 | 9 | 10 |
| 90% | MLR | 6 | 14 | 10 | 1 | 4 | 5 |
| | STAMPA | 12 | 17 | 18 | 2 | 6 | 6 |
| | LR | 48 | 52 | 58 | 8 | 22 | 35 |

**Table 4.3.** The comparison of MLR's and STAMPA's prediction accuracy and running time by using the number of tags (2, 5, 10, 15, 20, 25) on region ENr123 (A) and ENm010 (B) from 2 population: Han Chinese (HCB) and Japanese (JRT). Total number of SNPs in each dataset is in the parenthesis.

| Data | Methods | Measure | 2 | 5 | 10 | 15 | 20 | 25 |
|---|---|---|---|---|---|---|---|---|
| | MLR | accuracy | 0.803 | 0.928 | 0.981 | 0.992 | 0.998 | 0.999 |
| (A) | STAMPA | accuracy | 0.744 | 0.903 | 0.937 | 0.952 | 0.960 | 0.969 |
| HCB | MLR | RT (s) | 0.247 | 0.633 | 1.893 | 3.798 | 6.345 | 9.357 |
| (63) | STAMPA | RT (s) | 4.109 | 4.136 | 4.180 | 4.267 | 4.385 | 4.425 |
| JPT | MLR | accuracy | 0.814 | 0.938 | 0.980 | 0.994 | 0.998 | 1 |
| (63) | STAMPA | accuracy | 0.792 | 0.909 | 0.953 | 0.968 | 0.981 | 0.986 |
| | MLR | accuracy | 0.935 | 0.955 | 0.968 | 0.979 | 0.989 | 0.995 |
| (B) | STAMPA | accuracy | 0.895 | 0.938 | 0.956 | 0.960 | 0.966 | 0.969 |
| HCB | MLR | RT (s) | 0.763 | 1.388 | 3.978 | 10.308 | 13.345 | 15.357 |
| (123) | STAMPA | RT (s) | 3.451 | 3.462 | 3.652 | 3.655 | 3.852 | 4.425 |
| JPT | MLR | accuracy | 0.814 | 0.938 | 0.980 | 0.994 | 0.998 | 1 |
| (123) | STAMPA | accuracy | 0.792 | 0.909 | 0.953 | 0.968 | 0.981 | 0.986 |

STEAP, TRPM8, and 5q31 data will be 79.36%, 72.53%, and 63.57%, respectively. MLR first predicts each SNP as 0 and then gets even higher prediction accuracy when it uses a single tag while STAMPA requires at least two tags for prediction. Table 4.4 compares MLR with STAMPA on prediction accuracy and running time on 4 HapMap datasets.

According to the regression model (1), the tags which are more correlated with the predicted SNP have larger regression coefficients and, therefore, will contribute more to predicting the SNP. For example, for 7 ENCODE regions [5] and $k = 10$ tags, the tag with the largest regression coefficient ($\approx 0.82$ on average) has average correlation 0.61 with the predicted SNP, the tag with second largest regression coefficient has average correlation 0.28 and so on. Averaged over all considered real datasets, the correlation between regression coefficients and tag-to-SNP correlations is 0.96. When MLR-tagging is applied to data containing both high- and low-LD regions, the high-LD region have always small number of tags since tags in the low-LD region do not correlate with SNPs in the high-LD region and, therefore, do not contribute to high-LD SNP prediction. For ENm010JRT dataset containing 11 SNPs in the high-LD region and 94 SNPs in the low-LD region, only 2 tags are chosen in the high-LD region out of total $k = 59$ tags.

For maximizing statistical covering, each (non-tag) SNP-column $s$ is predicted with the MLR prediction algorithm. We say that the SNP-column $s$ is counted as *statistically covered* if squared correlation $R^2$ between the predicted SNP-column $s'$ and given SNP-column $s$ is at least 0.8. In Table 4.4, the first two rows show the correlation of prediction accuracy and number of statistically covered SNPs. The third row shows that it is slightly better to use the correct objective (i.e., statistical covering) rather than prediction accuracy in order to maximize the number of statistically covered SNPs. Table 4.5 shows that MLR/STA uses on average 30% fewer tags than IdSelect [16] for statistical covering all SNPs.

**Table 4.4.** The quality of MLR/STA on Daly et al. [26] data with two different tagging objectives over different number of tag SNPs.

| objective | | number of tag SNPs | | | | | | |
|---|---|---|---|---|---|---|---|---|
| of tagging | | 0 | 1 | 2 | 4 | 6 | 8 | 10 |
| SNP prediction | prediction accuracy, % | 61.54 | 81.35 | 83.94 | 88.65 | 91.11 | 92.96 | 93.89 |
| SNP prediction | # of SNPs covered | 0 | 10 | 16 | 36 | 47 | 53 | 59 |
| statistical covering | # of SNPs covered | 0 | 11 | 24 | 38 | 50 | 54 | 61 |

**Table 4.5.** The number of tag SNPs for statistical covering of all SNPs required by three methods: MLR/STA with prediction objective, MLR/STA with statistical covering objective, and IdSelect [16].

| Algorithm | ENm013 | ENr112 | ENr113 | STEAP | TRPM8 | 5q31 |
|---|---|---|---|---|---|---|
| MLR (prediction) | 56 | 82 | 106 | 13 | 46 | 44 |
| MLR (statistical covering) | 51 | 71 | 85 | 11 | 41 | 41 |
| IdSelect | 71 | 122 | 132 | 16 | 53 | 51 |

### 4.4.5 MLR-tagging Software

Here, we describe our tagging software base on multiple linear regression. MLRtagging software package implements a novel genotype tagging method based on multiple linear regression (MLR) analysis. This software can be used for tag selection and genotype prediction. The stepwise tag selection algorithm (MLRsta) selects positions of the given number of tags based on a genotype sample population. The MLR SNP prediction algorithm (MLRprediction) predicts a complete genotype based on the values of its tag SNPs, tag positions among all SNPs, and a sample of complete genotypes.

Downloading and Installing All relevant files including this pdf file are included in the tar files: available at http://alla.cs.gsu.edu/ software/tagging. Download this tar file to your machine then extract the files from the archive.

tar -xvf MLRtagging.tar Currently, there is only Linux version available.

The package contains the following files:

1. taggingReadme.pdf: Readme file

2. MLRsta: Binary code for tag selection

3. MLRprediction: A Binary code for SNP prediction

4. genoInput.txt: Sample input of a genotype population sample: 129 offspring genotypes each with 103 SNPs from Daly et al.[26]

5. tagGenoIndividual.txt: Sample input of unknown genotype with typed tags

6. tagFile.txt: Sample input of tag positions

***For running MLRsta:***

type ./MLRsta genoInput.txt 2 tagFile.txt ”

First parameter = the file name of a genotype sample population ”

Second parameter = desired number of tags k ”

Third parameter = the name of output tag file (it contains selected k tag positions)

***For running MLRprediction:***

type “./MLRprediction genoInput.txt tagFile.txt tagGenoIndividual.txt G fullIndividual.txt ”

First parameter = “the file name of a genotype sample population ”

Second parameter = “the name of input tag file (it contains selected k tag positions) ”

Third parameter = “the name of input file of a tag-restricted genotype with only tag SNP values ”

Fourth parameter = “the name of output file of a complete genotype

***File Formats:***

*genoInput.txt and tagGenoIndividual.txt contain the following lines:*

The number of genotypes

The number N of SNPs in each genotype

Description of data (can be empty)

The first genotype represented by a sequence of 0/1/2’s without gaps, 0 stands for homozygous major allele, 1 stands for homozygous minor allele, and 2 stands for heterozygous SNP.

.......

The last genotype

*tagFile.txt consists of k+3 lines:*

The number of tags

Description of data (can be empty)

Description of data (can be empty)

The position of the first tag (a number in the range from 0 to N-1, where N is the number of SNPs.)

.......

The last tag

## 4.5 Support Vector Machine SNP Prediction Method

We propose a new SNP prediction using a robust tool for classification – Support Vector Machine (SVM). For tag selection we use a fast stepwise tag selection algorithm. An extensive experimental study on various datasets including three regions from HapMap shows that the tag selection based on SVM SNP prediction can reach the same prediction accuracy as the methods of Halldorson et al. [42] on the LPL using significantly fewer tags. For example, our method reaches 90% non-tag SNP prediction accuracy using only three tags for Daly et al. [26] dataset with 103 SNPs. The proposed tagging method is also more accurate (but considerably slower) than multiple linear regression method of He et al. [45].

### 4.5.1 SVM Overview

SVM has recently attracted a lot of attention in bioinformatics research (see, e.g. [89]). This is because SVM produces very accurate results comparatively with other data mining approaches such as Neural Networks. The SVM method is a learning system which is developed by Vapnik and Cortes [94]. SVM is a powerful methodology for solving problems in nonlinear classification, function estimation and density estimation. The basic principle behind SVM is to find an optimal maximal margin separating hyperplane between two classes. The goal is to maximize the margin between the solid planes separating the two classes and at the same time permit the least amount of errors as possible. SVM can also be used in the case when the data is not linearly separable. In this case, the data is mapped to a high dimensional future space using a nonlinear function. When using SVM, the dot products (x,y) in the future space must be fed to the SVM, which can be computed through a positive definite kernel in the input space.

After given a training set (a set of pairs, input vector: features and target), SVM builds a model. This model is later applied to unknown test set where the model maps an input vector to +1 (positive class) or −1 (negative class) output target value.

SVMlight is an implementation of Vapnik's Support Vector Machine [94]. In this project, we have used $SVM^{light}$ software as a black box to do the prediction. The $SVM^{light}$ software has many features such as changing the kernel function and other parameters. We have used the Radial Basis Function (RBF) kernel in our project it is the default and recommended kernel function.

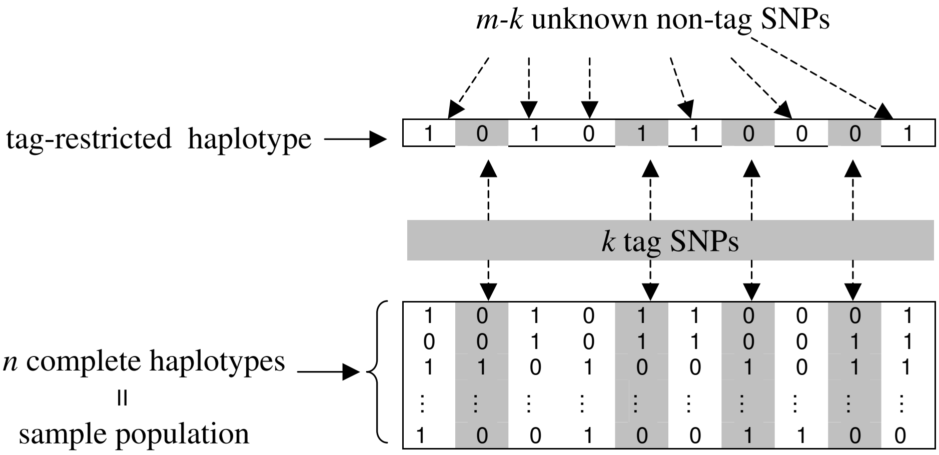
$$\exp(-\gamma * |u - v|^2)$$

For the trade-off between training error and margin, 0.05 is chosen (c value). Parameter gamma in RBF kernel was chosen as 0.1. These parameters were found by testing different values in our experiments. We used the same for all the experiments.

### 4.5.2 SVM Haplotype Tagging

This problem can be formulated as **Haplotype Tagging Problem** (see Figure 4.11). Given the full pattern of all haplotypes in a small population sample, find the minimum number of tag SNPs and the method for reconstructing each haplotype in the entire population from these tags.

This tagging problem formulation implicitly relies on a certain SNP prediction method. The corresponding **SNP prediction problem** is formulated as follows:



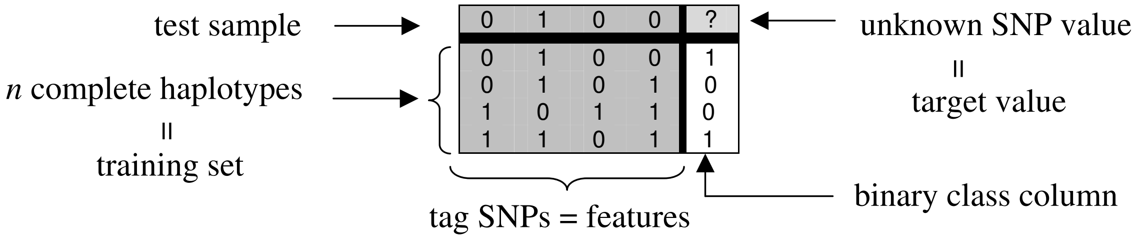
|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

**Figure 4.11.** Haplotype Tagging Problem. The shaded columns correspond to $k$ tag SNPs and the clear columns correspond to $m - k$ non-tag SNPs. The unknown $m - k$ non-tag SNP values in tag-restricted haplotype (top) are predicted based on the known $k$ tag values and the sample population of n complete haplotypes.

Given the values of $k$ tags of the individual $x$ with unknown SNP $s$ and $n$ individuals with $k$ tag SNP and known value of SNP $s$, find the value of $s$ in $x$.

In the SNP Prediction Problem, SVM builds a model after given n complete haplotypes as training set. Then when an unknown haplotype is given to SVM as a test sample, SVM is asked to predict the unknown SNP value (see Figure 4.12).



|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | ? |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |

**Figure 4.12.** The SNP Prediction Problem. Each haplotype with k tags in the training set belongs to the 0- or 1- class. These binary class values are given in the last column. For a given k tag-restricted haplotype (test sample), the unknown non-tag SNP in the right corner should be classified based on the known tag SNP values and training set.

### 4.5.3 Experimental Results

We apply our haplotype tagging algorithm (SVM/STA) to very well known haplotype datasets. These datasets are original genotype datasets, but we phased them to obtain haplotypes using GERBIL algorithms [37].

**Two gene Regions form HapMap.** Two gene regions STEAP and TRPM8 from 30 CEPH family trios are obtained from HapMap [5]. We took the HapMap SNPs that are spanned by the gene plus 10KB upstream and downstream. The number of SNPs genotyped in each gene region is 23 and 102 SNPs. We only use 60 haplotypes of parents.

**Chromosome 5q31.** The data set collected by Daly et al. [26] is derived from the 616 kilobase region of human Chromosome $5q31$ that may contain a genetic variant responsible for Crohn's disease by genotyping 103 SNPs for 129 trios. We only use 258 haplotypes of offsprings.

**LPL** The Clark et al. [21] data set consists of the haplotypes of 71 individuals typed over 88 SNPs in the human lipoprotein lipase (LPL) gene.

We apply leave-one-out cross-validation to evaluate the quality of the solution given by the tag SNP selection and prediction methods. One by one, each individual is removed from the sample. Then, tag SNPs are selected using only the remaining individuals. The "left out" individual is reconstructed based on its tag SNPs and the remaining individuals in the sample. The average number of errors in the reconstruction of all individuals is used as a measure of the overall prediction accuracy.

Table 4.6 presents the results of STA combined with SVM (SVM/STA) on leave-one-out experiments on the 3 haplotype datasets. Table 4.7 compares SVM/STA with multiple linear regression method (MLR) of He et al. [46] on the 3 haplotype datasets. The proposed tagging method is more accurate than multiple linear regression method of He et al. [45]. For example, for small number of tag SNPs, SVM/STA can obtain (up to 8%) better prediction accuracy than MLR with same number of tag SNPs.

But SVM/STA is considerably slower. Indeed, for 5q31 dataset, SVM/STA needs 3 hours to select 1 tag SNPs while MLR only needs 0.77 seconds[2].

| datasets (# of SNPs) | prediction accuracy % | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 80 | 85 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| 5q31 (103) | 1 | **1** | **3** | **3** | **4** | **5** | **6** | **8** | **10** | **22** | **42** | **51** |
| TRPM8 (101) | 1 | **1** | **2** | 5 | 5 | 6 | 7 | 8 | 10 | 15 | 15 | 24 |
| STEAP (22) | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | **2** | **2** | **2** | 2 |

**Table 4.6.** Leave-one-out tests are performed on 3 real haplotype datasets. The minimum number of tag SNPs needed to reach from 80% to 99% prediction accuracy is listed. The bold numbers indicate cases when the SVM/STA needs fewer tags than the MLR method of He et al. [45] for reaching same prediction accuracy.

| datasets (# of SNPs) | | methods | number of tag SNPs | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 4 | 6 | 8 | 10 |
| 5q31 (103) | prediction accuracy % | SVM/STA | 86.81 | 89.32 | 92.24 | 94.09 | 95.28 | 96.09 |
| | | MLR | 81.15 | 83.84 | 88.15 | 90.91 | 92.66 | 93.49 |
| | running time | SVM/STA | 3 hour | 5 hour | 11 hour | 16 hour | 18 hour | 1 day |
| | | MLR | 0.77 sec | 1.16 sec | 4.07 sec | 7.27 sec | 11.26 sec | 15.92 sec |
| TRPM8 (101) | prediction accuracy % | SVM/STA | 88.89 | 90.50 | 90.67 | 93.67 | 95.56 | 96.74 |
| | | MLR | 80.68 | 85.32 | 90.75 | 93.74 | 95.16 | 96.38 |
| | running time | SVM/STA | 1 hour | 2 hour | 5 hour | 9 hour | 16 hour | 23 hour |
| | | MLR | 0.357 sec | 0.787 sec | 1.895 sec | 3.376 sec | 5.181 sec | 7.373 sec |
| STEAP (22) | prediction accuracy % | SVM/STA | 94.02 | 98.18 | 99.68 | 99.73 | 99.79 | 99.80 |
| | | MLR | 90.79 | 96.16 | 99.13 | 99.71 | 99.78 | 99.78 |
| | running time | SVM/STA | 14 min | 27 min | 1 hour | 2 hour | 3 hour | 4 hour |
| | | MLR | 0.034 sec | 0.052 sec | 0.118 sec | 0.203 sec | 0.304 sec | 0.413 sec |

**Table 4.7.** The comparison of our proposed SVM/STA method and the MLR method of He et al. [45] over different number of tag SNPs.

We also compare SVM/STA with the methods of Halldorson et al. [42] and the method of He et al. [45] in leave-one-out tests on the LPL data set (see Figure 4.13). Note that the method of Halldorson et al. imputes a SNP based on the tag SNPs in the same neighborhood and in fact can be classified as a method for statistical coverage. If there is no tag SNPs in the neighborhood, then their method does not make any prediction. It is not surprising that it performs poorly for SNP prediction. The SVM/STA method reconstructs each SNP based on the values of *all* tag SNPs

[2] All experiments are performed on a computer with Intel Pentium 4, 3.06Ghz processor and 2 GB of RAM.

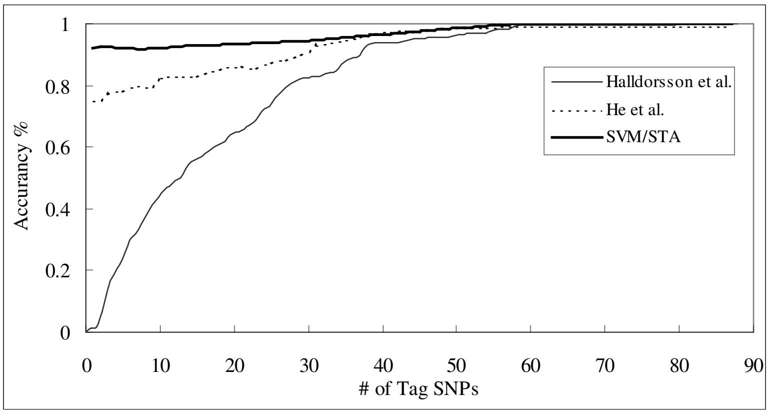


**Figure 4.13.** Comparison among three haplotype tagging method on LPL data: SVM/STA, Halldorson et al. [42], and He et al. [45] in a leave-one-out experiment. The x-axis shows the number of SNPs typed, and the y-axis shows the fraction of SNPs correctly imputed.

which may potentially be far away. On the LPL dataset, SVM/STA reaches, e.g., 90% accuracy using only one tag.

### 4.5.4 SVM-tagging Software

Here, we describe our tagging software base on multiple linear regression. SVM-tagging selects haplotype tag SNP using support vector machines. We first describe how to download, compile, and run SVMtagging package. Then we describe input and output formats.

Downloading and Installing All relevant files including this pdf file are included in the tar files: SVMtagging.tar - Linux version available at http://alla.cs.gsu.edu/ software.

Download this tar file to your machine then extract the files from the archive tar -xvf SVMtagging.tar

Compile leaveOneOutSVM.cpp: g++ -o leaveOneOutSVMleave OneOutSVM.cpp And checkExist.cpp: g++ -o checkExist checkExist.cpp

Download the SVMlight from http://svmlight.joachims.org/

Make sure perl is in $\sharp!/usr/bin/perl$

Make sure SVMlight and SVM-tagging package all are in same directory

***Running the Program For running SVM-tagging:***

type perl SVMtagging.perl hap.txt -g 10

- First parameter = the file name of a haplotype population sample

- Third parameter = desired number of tags K

The result tags will store in tag.txt.

***File Formats hap.txt contain the following lines:***

- The number of haplotypes

- The number N of SNPs in each haplotype

- Description of data (can be empty)

- The first haplotype represented by a sequence of 0/1's without gaps, 0 stands for major allele, 1 stands for minor allele.

.....

-The last haplotype

## 4.6 Application of Tagging to Disease Association Search

### 4.6.1 Multi-SNP to Disease Association

Recent improvement in accessibility of high-throughput DNA sequencing brought a great deal of attention to disease association and susceptibility studies. Successful genome-wide searches for disease-associated gene variations have been recently reported [86]. However, complex diseases can be caused by combinations of several unlinked gene variations. Disease association studies analyze genetic variation across exposed to a disease and healthy individuals. The difference between individual DNA sequences occurs at a single-base sites, in which more than one allele is observed across population. Such variations are called single nucleotide polymorphisms (SNPs). The number of simultaneously typed SNPs for association and linkage studies is reaching 250,000 for SNP Mapping Arrays [1]. High density maps of SNPs as well as massive

DNA data with large number of individuals and number of SNPs become publicly available [5]. Diploid organisms, like human, have two near-identical copies of each chromosome. Most genotyping techniques (e.g., SNP Mapping Arrays [1]) do not provide separate SNP sequences (*haplotypes*) for each of the two chromosomes. Instead, they provide SNP sequences (*genotypes*) representing mixtures of two haplotypes – each site is defined by an unordered pair of allele readings, one from each haplotype – while haplotypes are computationally inferred from genotypes [12]. The disease association study analyze data given as genotypes or haplotypes with disease status.

Disease association analysis searches for a SNP with frequency among exposed individuals considerably higher than among unexposed individuals. Only statistically significant SNPs (whose frequncy distribution has p-value less than 0.05) are reported. Successful as well as unsuccessful searches for SNPs with statistically significant association have been recently reported for different diseases and different suspected human genome regions (see e.g. [22]). Unfortunately, reported findings are frequently not reproducible on different populations. It is believed that this happens because the p-values are unadjusted to multiple testing – indeed, if the reported SNP is found among 100 SNPs then the probability that the SNP is associated with a disease by mere chance becomes roughly 100 times larger.

Since complex common diseases can be caused by multi-loci interactions two-loci analysis can be more powerful than traditional one-by-one SNP association analysis [67]. Multi-loci analysis is expected to find even deeper disease-associated interactions, therefore, we suggest to search for disease-associated multi-SNP combinations in the genotype/haplotype data.

An exhaustive search among multi-SNP combinations usually is infeasible even for small number of SNPs let alone the genome-wide studies. In order to handle data with huge number of SNPs we extract informative (indexing) SNPs that can be used for lossless reconstructing of all other SNPs using multiple linear regression based method [46]. However, exhaustive searching for all possible SNP combinations is still very slow. The main contribution here is a novel combinatorial method for finding disease-associated multi-SNP combinations applied to index SNPs.

Here we first propose to extract informative (indexing) SNPs that can be used for reconstructing of all SNPs almost without loss [46]. In the reduced set of SNPs, we propose to apply a novel combinatorial method for finding disease-associated multi-SNP combinations. Our experimental study shows that the proposed methods are able to find multi-SNP combinations whose disease association is statistically significant even after multiple testing adjustment. For Daly et al [26] data we found a few unphased multi-SNP combinations associated with Crohn's disease with multiple testing adjusted p-value below 0.05 while no single SNP or pair of SNPs show any significant association. For Ueda et al [93] data we found a few new unphased and phased multi-SNP combinations associated with autoimmune disorder.

### 4.6.2 Problem Formulation

In this section we formally describe the search of statistically significant disease-associated multi-SNP combinations.

Our data is a set of $n$ individuals described by values of $m$ SNPs and its disease status. When individuals are represented by genotypes then SNP values belong to $\{0, 1, 2\}$, where 0's and 1's denote homozygous sites with major allele and minor allele, respectively, and 2's stand for heterozygous sites. When individuals are represented by haplotypes then SNP values belong to $\{0, 1\}$, where 0's and 1's denote major allele and minor allele, respectively.

A *multi-SNP combination* $C$ is given by a subset of SNPs $snp(C)$ ($snp(C)$ is a subset of the set of all $m$ SNPs) and their values. The subset of individuals whose values match values of $C$ on the SNPs from $snp(C)$ is denoted $set(C)$ and its size is denoted

$size(C)$ . The set $set(C)$ is partitioned into two subsets: $exposed(C)$ consisting of exposed individuals and $unexposed(C)$ consisting of unexposed individuals.

We are interested in the probability that $set(C)$ is partitioned into $exposed(C)$ and $unexposed(C)$ by mere chance. According to binomial distribution, the p-value equals:

$$p(C) = \sum_{k=0}^{exposed(C)} \binom{n}{k} q^k (1-q)^{size(C)-k} \tag{4.5}$$

where $q = \frac{\#ofexposed}{n}$ is the probability of being exposed. The multi-SNP combination $C$ is statistically significant if the $p(C) < 0.05$.

The p-value computed by formula 4.5 is correct only if a single multi-SNP combination is tested. Since statistically significant multi-SNP combinations are searched among many such combinations, the computed p-value requires adjustment for multiple testing. The standard Bonferroni correction adjusts p-value by multiplying it by the number of the number of tests, i.e., number of tested multi-SNP combinations. However, the Bonferroni correction is overly pessimistic, e.g., for finding one significant SNP among 100 we should multiply its p-value by 100; as a result, SNP should have $p < 0.0005$ in order to be statistically significant. Similarly this factor grows to $10^4$ for 2-SNP combinations. Instead, we compute multiple testing adjustment using more accurate but computationally extensive randomization method. $10^4$ times we repeat the following: (1) randomize the status of individuals (by random swapping) and (2) find the 500th smallest p-value of multi-SNP combinations. This p-value corresponds to the multiple testing adjusted $p = 0.05$.

Formally the searching problem is as follows

**Disease-Associated Multi-SNP Combination Search.** Given a population of $n$ genotypes (or haplotypes) each containing values of $m$ SNPs and disease status (exposed or unexposed), find all multi-SNP combinations with multiple-testing adjusted p-value of the frequency distribution below 0.05.

### 4.6.3 Searching Methods for Disease Association

In this section we first discuss the exhaustive search for the multi-SNP combinations. Next, we briefly describe multiple linear regression method [46] for extracting informative index SNPs. Then we propose the new combinatorial search algorithm and its faster implementation.

**Exhaustive Search.** The search for disease-associated multi-SNP combinations among all possible combinations can be done by the following *exhaustive search*. In order to find a multi-SNP combination with the p-value of the frequency distribution below 0.05, we should check all one-SNP, two-SNP, ..., m-SNP combinations. The checking procedure takes $O(n\sum_{k=1}^{m}\binom{m}{k}3^k)$ runtime for unphased combinations since there are 3 possible SNP values $\{0, 1, 2\}$. Similarly, for phased combination, the runtime is $O(n\sum_{k=1}^{m}\binom{m}{k}2^k)$ since there only 2 possible SNP values. The exhaustive search is infeasible even for small number of SNPs and we limit ourselves with the small number of SNPs, i.e., instead of searching all multi-SNP combinations, we search only containing at most $k = 1, 2, 3$ SNPs. We refer to $k$ as *search level* of exhaustive search.

**Indexing with MLR Tagging Method.** In order to reduce the runtime of exhaustive search, we propose to decrease the size the input data set by extracting informative SNPs (further referred as *indexing SNPs*) from which one can reconstruct all other SNPs. In our experiments we use multiple linear regression tagging method of [46]. However, there is a tradeoff between the number of chosen SNPs and quality of reconstruction. Our strategy is to chose maximum number of index SNPs that still result in the reasonable runtime of the exhaustive search.

**Combinatorial Search.** Here we propose the following search method for disease-associated multi-SNP combinations. It can find a combinations with large number of SNPs even for small search levels.

Given a multi-SNP combination $C$, sometimes one can decrease the the frequency of $C$ among unexposed population by adding SNPs to the $snp(C)$ which have the same value on all individuals from $exposed(C)$. Such addition will not affect $exposed(C)$ but may reduce $unexposed(C)$.

Formally, a multi-SNP combination $C'$ is an *exposed-closure* of multi-SNP combination $C$, if $exposed(C') = exposed(C)$ and the size of $unexposed(C')$ is minimized. Two multi-SNP combinations $C$ and $C'$ are equivalent if $exposed(C) = exposed(C')$ and $unexposed(C) = unexposed(C')$. Equivalent multi-SNP combinations can not be distinguished, and we will represent the equivalence class by the multi-SNP combination with the maximum number of specified SNPs. This representative $C$ can be efficiently found by incorporating into $snp(C)$ *all* SNPs that have the same value across all individuals in $unexposed(C)$.

The proposed combinatorial search finds the best p-value of the exposed-closure of each single-SNP, after that it searches for the best p-value among exposed-closure of all 2-SNP combinations and so on. The procedure stops after all exposed-closure of all k-SNP combinations ($k < m$) are checked. The corresponding *search level* is the number of SNPs selected for exposed-closuring, e.g., on the level 2 of searching combinatorial search will test exposed-closure of all 2-SNP combinations for an association with a disease. Because of the exposed-closure, for the same level of searching combinatorial search finds better association than exhaustive search. However, proposed combinatorial search is as slow as exhaustive search.

**Speed-up of Combinatorial Search.** A faster implementation of this method avoids checking multi-SNP combinations which are not (and cannot lead to) statistically significant ones. Formally, a multi-SNP combination $C$ is called an *intersection* of multi-SNP combination $C_1$ and $C_2$ if $exposed(C) = exposed(C_1) \cap exposed(C_2)$ and $unexposed(C)$ is minimized. A multi-SNP combination $C$ is called *trivial* if its unadjusted p-value is larger than 0.05 even if the set $exposed(C)$ would be empty. Note that intersection of a trivial multi-SNP combination with another is trivial.

A faster implementation of the combinatorial search is as follows:

1. Compute a set $G_1$ of all 1-SNP exposed-closed multi-SNP combinations, exclude trivial combinations.

2. Compute sets $G_k$ of all pairwise intersections of the multi-SNP combinations from $G_{k-1}$, exclude trivial combinations and already existing in $G_1 \bigcup G_2 \bigcup .. \bigcup G_{k-1}$, $k = 2..N$.

3. For each $G_k$ output multi-SNP combinations whose unadjusted $p < 0.05$.

Still, in order to find all multi-SNP combinations associated with a disease we have to check all possible SNP combinations with all possible SNP values. Our searching approach is also computationally intensive and step 2 from the algorithm can generate an exponential number of multi-SNP combinations. However, exposed-closure avoids generation and checking of non-significant multi-SNP combinations. Additionally, removing of trivial multi-SNP combinations at each iteration of step 2 considerably reduces the number of newly generated multi-SNP combinations. For example, for search level 2 our method is faster than level-2 exhaustive search, and returns all possible disease-associated 2-SNP combinations as well as the set of multi-SNP combinations obtained by exposed-closure of 1- or 2-SNP combinations. In conclusion, proposed method is more efficient than the exhaustive search and can find multi-SNP combinations associated with a disease on small search levels.

### 4.6.4 Experimental Results

**Data Sets.** The data set Daly *et al* [26] is derived from the 616 kilobase region of human Chromosome $5q31$ that may contain a genetic variant responsible for Crohn's disease by genotyping 103 SNPs for 129 trios. All offspring belong to the case popu-

**Table 4.8.** Comparison of four methods for searching disease-associated multi-SNPs combinations.

| Search level | Search method | MT-unadjusted p corresponding to adjusted p=0.05 | SNP combination with minimum p-value | | | Number of SNP combinations with MT-adjusted p<0.05 | runtime sec. |
|---|---|---|---|---|---|---|---|
| | | | exposed frequency | unexposed frequency | unadjusted p-value | | |
| **Unphased Daly et al [26]** | | | | | | | |
| 1 | ES | $1.6{\times}10^{-3}$ | 0.31 | 0.16 | $1.8{\times}10^{-3}$ | 0 | 0.9 |
| | IES(30) | $3.9{\times}10^{-3}$ | 0.30 | 0.16 | $4.7{\times}10^{-3}$ | 0 | 0.5 |
| | CS | $5.1{\times}10^{-5}$ | 0.30 | 0.11 | $2.0{\times}10^{-5}$ | 2 | 1.0 |
| | ICS(30) | $2.2{\times}10^{-3}$ | 0.30 | 0.14 | $4.6{\times}10^{-4}$ | 1 | 0.6 |
| 2 | ES | $1.9{\times}10^{-5}$ | 0.30 | 0.13 | $3.1{\times}10^{-4}$ | 0 | 15.0 |
| | IES(30) | $1.0{\times}10^{-4}$ | 0.31 | 0.14 | $4.4{\times}10^{-4}$ | 0 | 1.0 |
| | CS | $1.5{\times}10^{-6}$ | 0.17 | 0.02 | $6.5{\times}10^{-7}$ | 2 | 7.0 |
| | ICS(30) | $5.0{\times}10^{-5}$ | 0.17 | 0.04 | $3.7{\times}10^{-5}$ | 1 | 0.4 |
| **Unphased Ueda et al [93]** | | | | | | | |
| 1 | ES | $1.3{\times}10^{-3}$ | 0.43 | 0.28 | $1.1{\times}10^{-4}$ | 2 | 1.0 |
| | IES(30) | $3.1{\times}10^{-3}$ | 0.43 | 0.28 | $1.1{\times}10^{-4}$ | 4 | 0.6 |
| | CS | $1.8{\times}10^{-4}$ | 0.43 | 0.28 | $9.2{\times}10^{-5}$ | 2 | 1.1 |
| | ICS(30) | $1.6{\times}10^{-3}$ | 0.43 | 0.28 | $1.1{\times}10^{-4}$ | 4 | 0.6 |
| 2 | ES | $2.7{\times}10^{-6}$ | 0.25 | 0.12 | $1.5{\times}10^{-6}$ | 2 | 30.0 |
| | IES(30) | $8.0{\times}10^{-5}$ | 0.25 | 0.12 | $1.5{\times}10^{-6}$ | 9 | 3.0 |
| | CS | $1.1{\times}10^{-6}$ | 0.16 | 0.06 | $8.5{\times}10^{-7}$ | 3 | 20.0 |
| | ICS(30) | $4.7{\times}10^{-5}$ | 0.25 | 0.12 | $1.1{\times}10^{-6}$ | 10 | 1.0 |
| **Phased Daly et al [26]** | | | | | | | |
| 1 | ES | $2.4{\times}10^{-3}$ | 0.52 | 0.40 | $9.7{\times}10^{-3}$ | 0 | 1.0 |
| | IES(30) | $7.2{\times}10^{-3}$ | 0.52 | 0.41 | $1.6{\times}10^{-2}$ | 0 | 0.6 |
| | CS | $1.3{\times}10^{-4}$ | 0.52 | 0.36 | $4.3{\times}10^{-4}$ | 0 | 1.1 |
| | ICS(30) | $1.6{\times}10^{-2}$ | 0.52 | 0.40 | $1.0{\times}10^{-2}$ | 1 | 0.7 |
| 2 | ES | $3.0{\times}10^{-5}$ | 0.05 | 0.01 | $1.4{\times}10^{-3}$ | 0 | 23.0 |
| | IES(30) | $1.7{\times}10^{-4}$ | 0.55 | 0.42 | $5.5{\times}10^{-3}$ | 0 | 3.0 |
| | CS | $7.0{\times}10^{-7}$ | 0.48 | 0.30 | $5.9{\times}10^{-5}$ | 0 | 17.0 |
| | ICS(30) | $5.8{\times}10^{-5}$ | 0.48 | 0.35 | $3.1{\times}10^{-3}$ | 0 | 1.0 |
| **Phased Ueda et al [93]** | | | | | | | |
| 1 | ES | $9.2{\times}10^{-4}$ | 0.65 | 0.53 | $3.2{\times}10^{-4}$ | 2 | 6.0 |
| | IES(30) | $5.3{\times}10^{-3}$ | 0.66 | 0.55 | $1.4{\times}10^{-3}$ | 2 | 2.0 |
| | CS | $8.3{\times}10^{-4}$ | 0.37 | 0.28 | $2.9{\times}10^{-4}$ | 5 | 6.2 |
| | ICS(30) | $7.4{\times}10^{-2}$ | 0.66 | 0.55 | $1.4{\times}10^{-3}$ | 10 | 2.1 |
| 2 | ES | $2.1{\times}10^{-6}$ | 0.17 | 0.09 | $6.8{\times}10^{-7}$ | 2 | 173.0 |
| | IES(30) | $1.7{\times}10^{-4}$ | 0.19 | 0.12 | $3.7{\times}10^{-5}$ | 2 | 16.0 |
| | CS | $5.0{\times}10^{-7}$ | 0.02 | 0.00 | $1.6{\times}10^{-8}$ | 8 | 75.0 |
| | ICS(30) | $9.5{\times}10^{-5}$ | 0.19 | 0.12 | $3.0{\times}10^{-5}$ | 2 | 5.7 |

lation, while almost all parents belong to the control population. In entire data, there are 144 case and 243 control individuals.

The data set of Ueda *et al* [93] are sequenced from 330kb of human DNA containing gene CD28, CTLA4 and ICONS which are proved related to autoimmune disorder. A total of 108 SNPs were genotyped in 384 cases of autoimmune disorder and 652 controls.

The both datasets have been phased using 2SNP software [12]. The missing data (16% in [26] and 10% in [93]) have been imputed in genotypes from the resulted haplotypes. For each genotype dataset, we have also created corresponding haplotype dataset in which each individual is represented by a haplotype with the disease status inherited from the corresponding individual genotype.

**Search Methods.** We have compared the following 4 methods for search disease-associated multi-SNP combinations.

- Exhaustive Search (**ES**);
- Indexed Exhaustive Search (**IES(30)**): exhaustive search on the indexed datasets obtained by extracting 30 indexed SNPs with MLR based tagging method [**?**];
- Combinatorial Search (**CS**);
- Indexed Combinatorial Search (**ICS(30)**): combinatorial search on the indexed datasets obtained by extracting 30 indexed SNPs with MLR based tagging method [46].

Each of these methods have been applied only to the search levels of 1 and 2 even when only 30 indexed SNPs are used. All experiments were ran on Processor Pentium 4 3.2Ghz, RAM 2Gb, OS Linux – the runtime is given in the last column of Table 4.8.

**Performance Quality.** The quality of searching methods is compared by the number of found statistically significant multi-SNP combinations (see the 7th column of Table

4.8). Since statistical significance should be adjusted to multiple testing, we report for each method and data set the 0.05 threshold adjusted for multiple testing (this threshold is computed by randomization and given in the third column of Table 4.6). In the 4th, 5th and 6th columns, we give the frequencies of the best multi-SNP combination among exposed and unexposed population and the unadjusted p-value, respectively.

**Discussion.** Comparing indexed counterparts with exhaustive and combinatorial searches shows that indexing is quite successful. Indeed, indexed search finds the same multi-SNP combinations as non-indexed search but it is much faster and its multiple-testing adjusted 0.05-threshold is higher and easier to meet.

Comparing combinatorial searches with the exhaustive counterparts is advantageous to the former. Indeed, for unphased data [26] the exhaustive search on the first and second search levels is unsuccessful while the combinatorial search finds several statistically significant multi-SNP combinations. Similarly, for unphased and phased data of [93] the combinatorial search found much more statistically significant multi-SNP combinations then the exhaustive search.

# CHAPTER 5

# DISEASE SUSCEPTIBILITY PREDICTION

The final goal of genetic epidimiology is to identify gene variations or, in general, haplotypes and genotypes which are susceptible to a particular disease. Statistical association analysis usually results in claims that a presence of a given SNP considerably increases the risk of a certain disease which are of limited use for disease susceptibility because of the following two reasons. Statistical methods may lead to a quite plausible assertion that each case of complex diseases may have a unique chain of genetic as well as environmental elements [22]. Chapter 5 of this dissertation explores possibility of applying combinatorial methods to known case/control studies with the hope to reliably (to certain extent) predict disease susceptibility.

Section 5.1 introduces the disease susceptibility prediction problem. Next in sectino 5.2, we describe how remove redundant SNPs and propose several prediction algorithms which are mostly based on combinatorial optimization in Section 5.3. Then we describe the leave-one-out test for comparative estimation of prediction quality of a discrimination algorithm as well as bootstrapping strategy for estimation of significance of obtained results in Section 5.4..

We apply proposed methods to two data sets. The first data set consists of case/control study of Crohn's disease [26] of 129 family trios. The other set for autoimmune disorder [93] consists of 1036 unrelated case/control individuals. We achieved correct prediction rate of 77.28% and 64.77%, respectively. After applying bootstrapping we obtain with 95% confidence the correct prediction rate of 75.38% for Crohn's disease. We have also performed a *Monte-Carlo test* by running our methods on Crohn's disease's data with randomly swapped case/control markers. The average prediction rate falls to 50% for all proposed methods. This confirms predominating genetic susceptibility of Crohn's disease [7], high association of the chosen haplotype region with Crohn's disease as well as capabilities of the proposed methods to detect such susceptibility.

## 5.1 Introduction

### 5.1.1 Previous Work

Recent improvement in accessibility of high-throughput genotyping brought a great deal of attention to disease association and susceptibility studies [98]. High density maps of single nucleotide polymorphism (SNPs) [5] as well as massive genotype data with large number of individuals and number of SNPs become publicly available [26, 37]. By now most of analysis of the new data is undertaken in statistics community [22, 99]. We pursue a different line of attack on disease susceptibility adhered to computational community with an emphasis on designing rather than analytical methodology.

The main goal of disease susceptibility analysis is to identify gene variations or, in general, haplotypes and genotypes which are susceptible to a particular disease. If complex diseases are affected by multiple genes, the traditional direct statistical association so far is unsatisfactory and arguably is not applicable since it mostly relies on an assumption that the disease is caused by a single Mendelian gene [22], but some complex diseases, such as psychiatric disorders, are characterized by a non mendelian, multifactorial genetic contribution with a number of susceptible genes interacting with each other[11, 68]. Statistical association analysis usually results in claims that a presence of a given SNP considerably increases the risk of a certain disease which are of limited use for disease susceptibility because of the following two reasons. Firstly, it is difficult to derive a meaningful conclusion in case of a disease probability being, e.g., 10 in a million and the resulted increased probability being 20 in a million - such a negligible absolute probability increase is unreliable. Secondly, the SNPs susceptibility to complex diseases are usually linked and do not, therefore, have an increased cumulative impact as it would be expected from the independent SNPs.

There is a huge literature on classification and prediction algorithms and general approaches such as neural networks [84] and SVM [94], but most previous research are based on statistics.

Falk et al. [31] proposed an alternative use of relative risk (RR) statistic in calculating the risk of disease in the presence of particular antigens or phenotypes. This alternative uses, as the control sample, the parental antigens or haplotypes not present in the affected child. The formulation of a haplotype relative risk (HRR) thus eliminates the problems of sampling from the same homogeneous population to form both the disease sample and an appropriate control. In families selected through a single affected individual, where transmission of the four parental haplotypes can be followed unambiguously, the mathematical expectation of the HRR is identical to that of the RR. Since the sample formed from the 'non-affected' parental haplotypes is clearly from the same population as the disease sample, the HRR thus provides a reliable alternative to the RR. A further advantage obtains when family data are being collected as part of a study since the control sample is then automatically contained in the family material. Data from studies of patients with insulin dependent diabetes mellitus (IDDM) are used to obtain an estimate of the risk to those with HLA antigens or phenotypes associated with IDDM using the HRR statistic. A comparison of the HRR's and RR's for these data is also presented.

One major problem in studying an association between a marker locus and a disease is the selection of an appropriate group of controls. However, this problem of population stratification can be circumvented in a quite elegant manner by family-based methods. The haplotype-relative-risk (HRR) method, which samples nuclear families with a single affected child and uses the parental haplotypes not transmitted to that child as a control individual, represents such a method for estimating the relative risk of a marker phenotype. In the special case of a recessive disease, it was already known that the equivalence of the HRR method with the classical relative risk (RR) obtained from independent samples holds only if the probability theta of a recombination between marker and disease locus is zero.

Knapp et al. [59] extend this result to an arbitrary mode of inheritance. Furthermore, they compare the distribution of the estimators for HRR and RR and show that, in the case of a positive linkage disequilibrium between a marker and disease allele, the distribution of the estimator for HRR is (stochastically) smaller than that for RR, irrespective of the recombination fraction. The practical implication of this result is that, for the HRR method, there is no tendency to give unduly high risk estimators.

Both methods addressed above are based on statistic analysis of haplotypes: haplotype-relative-risk (HRR). There are many classification methods are applied also.

Listgarten et al. [66] introduced predictive models for breast cancer susceptibility from multiple SNPs. To test the influence of genetic polymorphisms on breast cancer risk, they have measured 98 SNPs distributed over 45 genes of potential relevance to breast cancer etiology in 174 patients and have compared these with matched normal controls. Using machine learning techniques such as support vector machines (SVMs), decision trees, and naive Bayes, they identified a subset of three SNPs as key discriminators between breast cancer and controls. IN their experiments, the SVMs performed maximally among predictive models, achieving 69% predictive power in distinguishing between the two groups, compared with a 50% baseline predictive power obtained from the data after repeated random permutation of class labels (individuals with cancer or controls). However, the simpler naive Bayes model as well as the decision tree model performed quite similarly to the SVM. They have shown that multiple SNP sites from different genes over distant parts of the genome are better at identifying breast cancer patients than any one SNP alone.

Tomita et al. [92] introduced the Criterion of Detecting Personal Group (CDPG) for extracting risk factor candidates(RFCs). RFCs are extracted using binomial test and random permutation tests. CDPG performs exhaustive combination analysis using case/control data and assumes the appearance of case and control subjects belonging to a certain rule as a series of Bernoulli trials, where two possible outcomes are case and control subjects with some probabilities.

The observed weakness of statistical methods may lead to a quite plausible assertion that each case of complex diseases may have a unique chain of genetic as well as environmental elements [22]. On the contrary, this study tries to assess accumulated information using combinatorial tools hoping that there exist certain combinatorial dependencies in haplotypes which are scattered all over lengthy sequences and are difficult to recover and, therefore, have not yet been (statistically) analyzed. We explore possibility of applying combinatorial methods to known case/control studies with the hope to reliably (to certain extent) predict disease susceptibility.

### 5.1.2 Problem Formulation

The general disease prediction algorithm is described as follows. Data sets have $n$ genotypes and each has $m$ SNPs. The ***input*** for genotype-based prediction algorithms includes:

(G1) Training genotype set $g_i = (g_{i,j}), i = 0, \dots, n-1, j = 1, \dots m$, $g_{i,j} \in \{0, 1, 2\}$

(G2) Disease status $s(g_i) \in \{-1, 1\}$, indicating if $g_i, i = 0, \dots, n-1$ , is in case (1) or in control (-1) , and

(G3) Testing genotype $g_n$ without any disease status.

The input for the haplotype-based prediction algorithms includes:

(H1) Training haplotype pair set $g_i = (h_{2i,j}, h_{2i+1,j}), i = 0, \ldots n-1, j = 1, \ldots m$, $h_{i,j} \in \{0, 1\}$

(H2) Disease status $s(g_i) \in \{-1, 1\}$, indicating if $g_i, i = 0, \ldots n-1$, is in case (1) or in control (-1) , and

(H3) Testing haplotype pair $g_n = (h_{2n}, h_{2n+1})$ without disease status.

We will refer to the parts (G1-G2), resp. (H1-H2), of the input as *training set* and to the part (G3), respectively (H3) as the test case. The ***output*** of the both types of prediction algorithms is the disease status of the genotype $g_n$, i.e., $s(g_n)$.

Haplotype-based algorithms are supposed to more accurately take in account combinatorial structure of data but they suffer from the noise contributed by uncertainty in phasing of original genotype data. Our experiments show that adaptation of genotype-based algorithms to haplotypes only slightly affects accuracy of prediction indicating high quality of our phasing method.

### 5.1.3 Measures of Prediction Quality and Cross-validation Methods

To measure the quality of prediction methods, we need to measure deviation between the truly disease status and the result of predicted susceptibility, which can be regarded as measurement error. we will present the basic measures used in epidemiology to quantify accuracy of detection and classification methods. These measures can be applied to the detection of any entity, of course, whether it is a disorder, an exposure, or any characteristic.

Epidemiologists employ separate, complementary measures for the correct classification of cases and of controls. The basic measures are:

**Sensitivity**: the proportion of persons who have the disease who are correctly identified as cases.

**Specificity**: the proportion of people who do not have the disease who are correctly classified as controls.

The definitions of these two measures of validity are illustrated in the following contingency table.

| | | True Status | | | |
|---|---|---|---|---|---|
| | | + | - | | |
| Classified | + | a | b | a + b | Positive tests |
| status | - | c | d | c + d | Negative tests |
| Total | | a + c | b + d | | |
| | | Cases | Controls | | |

**Table 5.1.** Classification contingency table

In this table:

***Sensitivity*** (accuracy in classification of cases ) = a / (a + c)

***Specificity*** (accuracy in classification of controls) = d / (b + d)

Sometimes the following terms are used to refer to the four cells of the table:

a = True positive, TP - people with the disease who test positive

b = False positive, FP - people without the disease who test positive

c = False negative, FN - people with the disease who test negative

d = True negative, TN - people without the disease who test negative

From the table, we also can compute accuracy and risk rates:

$$\begin{aligned} accuracy &= \frac{a+d}{a+b+c+d} \\ RiskRate &= \frac{\frac{a}{a+b}}{\frac{c}{c+d}} \end{aligned}$$

Sensitivity is the ability to correctly detect a disease. Specificity is the ability to avoid calling normal as disease. Accuracy is the percent of the population that are correctly predicted. Risk rate is the measure of the risk factor.

Cross-validation and bootstrapping are both methods for estimating generalization error based on "resampling" [96, 28, 53, 80]. The resulting estimates of generalization error are used to evaluate models or algorithms. In k-fold cross-validation, you divide the data into k subsets of (approximately) equal size. You train the net k times, each time leaving out one of the subsets from training, but using only the omitted subset to compute whatever error criterion interests you [96]. If k equals the sample size, this is called "leave-one-out" cross-validation. "Leave-v-out" is a more elaborate and expensive version of cross-validation that involves leaving out all possible subsets of v cases. Cross-validation is quite different from the "split-sample". In the split-sample method, only a single subset (the validation set) is used to estimate the generalization error, instead of k different subsets; i.e., there is no "crossing". The distinction between cross-validation and split-sample validation is extremely important because cross-validation is markedly superior for small data sets. Cross-validation can be used simply to estimate the generalization error of a algorithm (prediction method), or it can be used for algorithm selection by choosing one of several algorithms that has the smallest estimated generalization error. For example, you might use cross-validation to choose the number of hidden units, or you could use cross-validation to choose a subset of the inputs (subset selection). A subset that contains all relevant inputs will be called a "good" subsets, while the subset that contains all relevant inputs but no others will be called the "best" subset. Note that subsets are "good" and "best" in an asymptotic sense (as the number of training cases goes to infinity). With a small training set, it is possible that a subset that is smaller than the "best" subset may provide better generalization error.

Leave-one-out cross-validation often works well for estimating generalization error so we decide to use it to compute four numbers in confusion matrix, TP, TN, FP and FN. We also apply Leave-many-out cross-validation on both data sets.

Bootstrapping seems to work better than cross-validation in many cases [28]. In the simplest form of bootstrapping, instead of repeatedly analyzing subsets of the data, you repeatedly analyze subsamples of the data. Each subsample is a random sample with replacement from the full sample. The number of subsample be used may vary from 50 to 2000. There are many more sophisticated bootstrap methods that can be used not only for estimating generalization error but also for estimating confidence bounds for network outputs [28]. For estimating generalization error in classification problems, the .632+ bootstrap (an improvement on the popular .632 bootstrap) is one of the currently favored methods that has the advantage of performing well even when there is severe overfitting.

We use bootstrapping to compute the 95% confidence interval for each measure.

We have applied leave-one-out cross-validation to evaluate the quality of susceptibility-predicting algorithms as follows. We predict the disease status of each genotype in the given data set by applying the susceptibility-predicting algorithm to the rest of the data which is regarded as the training set. Then we compare the predicted susceptibility with the actual disease status. We report the prediction rate separately for cases and controls as well as for the entire population.

For verification purposes we also perform the following Monte-Carlo random test (MCT). The original association of $n$ genotypes with markers is randomly scrambled, i.e., we repeatedly randomly swap case and control markers and then run the prediction algorithm. The expected prediction rate should be 50% corresponding to random data prediction. Such test confirms that the programming implementation does not rely on illegal information, and that the genotype data contain disease susceptibility.

The confidence level of obtained data is confirmed by bootstrapping. We have performed a specified number of random samplings with replacement from the original data and then run our algorithms on each of these samples. The reported prediction rate is the worst observed in 95% of samples of the resulted distribution.

Another way of bootstrapping is to randomly choose 20 cases and 200 controls as training set and predict others, and repeat these samplings 100 times reporting the average prediction rate.

## 5.2 Disease Tagging

### 5.2.1 Problem Formulation

Constructing a complete human haplotype map is helpful when associating complex diseases with their related SNPs. Unfortunately, the number of SNPs is very large and it is costly to sequence many individuals. Therefore, it is desirable to reduce the number of SNPs that should be sequenced to a small number of informative representatives called *tag SNPs.* Also, the tag SNP selection may reduce the noise introducing by irrelevant SNPs for disease susceptibility study. It is important to genotyping/analysis a limited number of suspicious SNPs. The problem can be formulated ad the following disease tagging problem.

The input for the minimal disease tagging problem consists of:

- $n$ genotypes with $k$ SNPs $g_i | i = 0, \ldots n$, $g_{i,k} \in \{0, 1, 2\}$ partitioned into classes.
- every $g$ in the same class has the same marker $m(g)$, e.g. if genotypes are partitioned into the sick/healthy classes, we create 0-1-markers $m(g_i) \in \{0, 1\}$, indicating if $g_i | i = 0, \ldots n$ is healthy (0) or sick (1).
- $n$ pairs of haplotypes $h_{2i}$ and $h_{2i+1}$ (phased genotype for $g_i | i = 0, \ldots n$), each haplotype $h_{i,k} \in \{0, 1\}$.

The input for the minimal disease tagging problem also can be defined as a genotype graph as follows:

*Genotype* graph $G_x = \{H, G\}$, where the vertices $H$ are distinct haplotypes from $\{h_i | i = 1, \ldots m\}$ and the edges $G$ are genotypes $\{g_i | i = 1, \ldots, n\}$ each connecting its two haplotypes. Color of edge $c(e_i) = m(g_i)$.

We can formulate the minimal disease tagging problem as follows:

**Minimal Disease Tagging Problem (MDTP)** Given a set of phased genotypes partitioned into groups (e.g., case/control and unknown, etc.), find minimal number of tag SNPs that distinguish any two genotypes from different groups, (i.e., there exists one haplotype belongs to one genotype but not the another).

### 5.2.2 Reduction to Set Covering Problem

The disease tagging problem is reducible to an instance of the set covering problem.

Any pair of genotypes $e = (g, g^{'})$ from different classes, i.e., $g \in C, g^{'} \in C^{'}$ and $C \cap C^{'} = \emptyset$, should be distinguished by tag SNPs. For the genotype $g$, its corresponding haplotype pair is ($u$, $v$) and $u < v$ (all haplotypes are sorted according to alphabetic order of haplotype string). For the genotype $g^{'}$, its corresponding haplotype pair is ($y$, $z$) and $y < z$.

Given an instance of disease tagging problem (**TI**), we construct an instance (X, F) of the set-covering problem (**SVI**) as follows:

- The *set* X consists of the elements, two of which are created for $e$ with $x = (uv|yz)$ and $x^{'} = (uz|vy)$.
- The *family* F consists of subsets of X. Number of subsets is equal to number of SNPs. Each subset $S$ *covers* a set of elements of $x = (uv|yz)$ if either $s_u \neq s_v$ or $s_y \neq s_z$, and $x^{'} = (uz|vy)$ if either $s_u \neq s_z$ or $s_v \neq s_y$, where ($s_u, s_v, s_y$,$s_z$) represent corresponding SNP values of haplotypes (u, v, y, z).

The tag SNP set S is solution of TI iff F(S) is the solution of SCI.

Proof:

(a) The tag SNP set X is solution of TI $\Rightarrow$ F(S) is the solution of SCI.

Case 1: If a single tag SNP $s$ distinguishes a pair $e = (g, g^{'})$ in such way that the SNP value of one haplotype is different from that of the other three, s covers two elements $(x, x^{'})$ of $e$. For example, if $s_u \neq (s_v = s_y = s_z)$, $s$ covers both $x$ for $s_u \neq s_y$, and $x^{'}$ for $s_u \neq s_z$.

Case 2: If a single tag SNP $s$ distinguishes a pair $e = (g, g^{'})$ in such way that the SNP values of two haplotypes from the same genotype are identical and the SNP value of haplotypes from one genotype is different from the another one, s covers two elements $(x, x^{'})$ of $e$. For example, if $(s_u = s_v) \neq (s_y = s_z)$, $s$ covers both $x$ for $s_u \neq s_y$, and $x^{'}$ for $s_u \neq s_z$.

Case 3: A pair $e = (g, g^{'})$ can be only distinguished by two SNPs $(s, s^{'})$ in such way that the two SNP values of the corresponding two haplotypes of one genotypes are $(00, 10)$ and the others are 01 and 11, $s$ and $s^{'}$ cover two elements $(x, x^{'})$ of $e$ together. For example, if $s_u s_u^{'} = 00, s_v s_v^{'} = 10, s_y s_y^{'} = 01$ and $s_z s_z^{'} = 11$, $s$ covers $x^{'}$ for $s_u \neq s_z$, and $s^{'}$ covers $x$ for $s_u^{'} \neq s_v^{'}$.

(b) F(S) is the solution of SCI $\Rightarrow$ the tag SNP set S is solution of TI.

If the element pair $(x, x^{'})$ of a genotype pair $e$ is covered by a single subset s, it satisfies that the SNP value of one haplotype is different from that of the other three, or the SNP values of two haplotypes from the same genotype are identical and the SNP value of haplotypes from one genotype is different from the another one. So, the selecting SNP is a tag SNP to distinguish $e$.

If the the element pair $(x, x^{'})$ of a genotype pair $e$ is covered by only two subset s and $s^{'}$, it satisfies that the two SNP values of the corresponding two haplotypes of one genotypes are $(00, 10)$ and the others are 01 and 11. So, the selecting two SNPs are tag SNPs to distinguish $e$.

### 5.2.3 Set Covering Greedy Algorithm

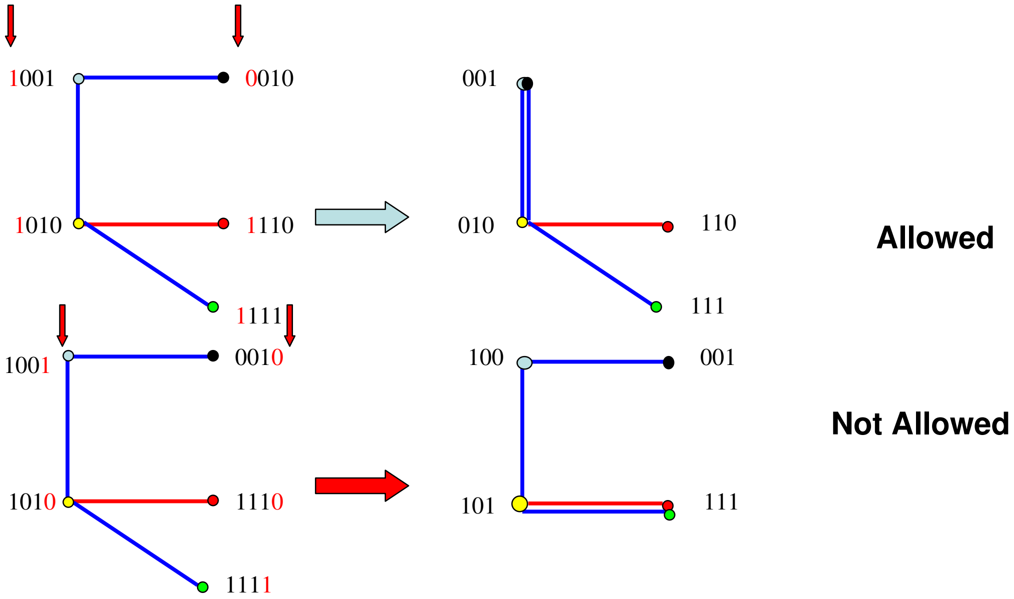


**Figure 5.1.** Set covering greedy algorithm for disease tagging

We apply greedy algorithm for deleting SNPS. We delete certain SNPs, keeping only tag SNPs. after deletion, vertices or edges may collapse. Our rule is that no red and blue edges are allowed to collapse after deleting SNPs. we greedily delete the SNPs which collapse most edges without violating the rule (see Figure 5.1

## 5.3 Prediction Algorithms for Disease Susceptibility

Below we describe the prediction algorithms which have been implemented and verified on both data sets. Although there is a huge literature on classification and prediction algorithms and general approaches such as neural networks [84], SVM [94] and classification trees [2], we decided to confine ourselves mostly to statistical and combinatorial algorithms. We first describe the statistics prediction method, then describe the graph-based method.

### 5.3.1 Statistics Methods

**Genotype Statistics.** This is a standard statistics method based on allele frequency. For each SNP $j = 1, \ldots, m$, we find the allele frequency $fd_j(0)$ of 0's in the case population. Similarly, for the same SNP $j$, we find the allele frequency $fd_j(1)$ of 1's and $fd_j(2)$ of 2's in the case population and the corresponding frequency in the control population $(fh_j(0), fh_j(1), fh_j(2))$. Then do next as follows.

$$f_j(0) = \ln \frac{fd_j(0)}{fh_j(0)}$$
$$f_j(1) = \ln \frac{fd_j(1)}{fh_j(1)}$$
$$f_j(2) = \ln \frac{fd_j(2)}{fh_j(2)}$$

For $g_n$, if $\sum_{j=1}^{m} f_j(g_{n,j})$ is greater than 0, then the output disease status $s(g_n)$ is set to 1 ($g_n$ is predicted to be in case population) and -1, otherwise.

**Haplotype Statistics.** This is a modification of the previous method in which we take in account fact that 2 means that the SNP is heterozygous. We replace genotype 0 with haplotype 00, 1 with 11 and 2 with 01 and follow the Genotype Statistics algorithm. As a result, new values can be expressed from the previous as follows.

$$\begin{aligned} fd_j(0) &= fd_j(0) + \frac{1}{2} fd_j(2) \\ fd_j(1) &= fd_j(1) + \frac{1}{2} fd_j(2) \\ fd_j(2) &= fd_j(0) fd_j(1) \end{aligned}$$

Similarly we compute $fh_j(0)$, $fh_j(1)$ and $fh_j(2)$ for the control population and $f_j(0)$,$f_j(1)$ and $f_j(2)$. If $\sum_{j=0}^{m} f_j(g_{n,j})$ is greater than 0, then the output marker $s(g_n)$ is set to 1 ($g_n$ is predicted to be in case population) and -1, otherwise.

### 5.3.2 Graph-based Prediction Methods

Our most successful methods are based on the following *genotype* graph $X = \{H, G\}$, where the vertices $H$ are distinct haplotypes from $\{h_i | i = 0, \ldots, 2n-1\}$ and the edges $G$ are genotypes $\{g_i | i = 0, \ldots, n-1\}$ each connecting its two haplotypes (vertices).

When applying graph heuristics to $X$, we found that it is necessary to increase the density of $X$. This can be achieved by dropping certain SNPs (or, equivalently, keeping only certain tag SNPs). Indeed, dropping a SNP may result in collapsing of certain vertices in $X$, i.e., different vertices become identical. Collapsing vertices may also result in collapsing certain edges (genotypes). A SNP dropping is not allowed if that results in collapsing edges from case and control populations, but collapsing of edges from the same population is allowed.

A simple greedy strategy consisting of (1) traversing all the SNPs and (2) dropping a SNP if it is allowed that will result in keeping a minimal subset of SNPs which do not collapse genotypes from opposite populations. Unfortunately, in the original graph $X$ we may already have collapsed edges from opposite populations - in fact, Daly *et al.* data contain such pair of genotypes. In this case we just remove the both genotypes in training set. Our experiments show that on average, we are left with 22 tag SNPs out of 103 for Daly *et al.* [26] data and 34 tag SNPs out of 108 for Ueda *et al.* [93] data.

After collapsing the graph $X$ we add the edge corresponding to the test-case genotype $g_n$. If the edge $g_n$ collapses with another edge $g_i$, then we set the predicted disease status $s(g_n) = s(g_i)$. Otherwise, we apply one of the following three methods for computing the disease status $s(g_n)$.

**First Neighbor.** $s(g_n)$ attains 1 if

$$\sum_{e \text{ adjacent to } g_n} s(e) > 0$$

and -1, otherwise. In other words, the predicted disease status is decided by voting among all adjacent edges.

**Second Neighbor.** $s(g_n)$ attains 1 if

$$\sum_{e \text{ adjacent to } g_n} \left( s(e) - \frac{\sum_{e' \text{ adjacent to } e} s(e')}{\delta(e)} \right) > 0$$

and -1, otherwise, where $\delta(e)$ is the number of edges adjacent to $e$. This method enhances the First Neighbor algorithm by taking in account the second neighbors.

**Haplotype Weighting.** This method assumes that certain haplotypes are susceptible to the disease while others are resistant to the disease. The genotype susceptibility is then assumed to be a sum of susceptibilities of its two haplotypes.

We want to assign a positive weight to susceptible haplotypes and the negative weight to resistant haplotypes such that for any control genotype the sum of weights of its haploptypes is negative and for any case genotype it is positive. We would also like to maximize the confidence of our weight assignment which can be measured by the absolute values of the genotype weights. In other words, we would like to maximize the sum[1] of absolute values of weights over all genotypes.

Formally, for each vertex $h_i$ of the graph $X$ we wish to assign the weight $p_i$,

$$-1 \leq p_i \leq 1 \qquad (5.1)$$

such that for any genotype-edge $e_{ij} = (h_i, h_j)$,

$$s(e_{ij})(p_i + p_j) \geq 0 \qquad (5.2)$$

and the total sum of absolute values of genotype weights is maximized

[1] Instead, we may maximize minimum absolute value over all genotype weights. Our experiments show that the results are quite similar for the both objectives.

$$\sum_{e_{ij}=(h_i,h_j)} s(e_{ij})(p_i + p_j) \tag{5.3}$$

The formulation (5.1-5.3) is a linear program which can be efficiently solved by a standard linear program solver such as CPLEX [3] or LPsolve [4].

## 5.4 Experimental Results

**Data Sets**The data set Daly *et al.* [26] is derived from the 616 kilobase region of human Chromosome 5$q$31 that may contain a genetic variant responsible for Crohn's disease by genotyping 103 SNPs for 129 trios. The missing data in genotypes is comparatively high (about 16%) – on average 10 SNPs per individual's genotype are missing. All offspring belong to the case population, while almost all parents belong to the control population. In entire data, there are 144 case and 243 control individuals.

The data set of Ueda *et al.* [93] are sequenced from 330kb of human DNA containing gene CD28, CTLA4 and ICONS which are proved related to autoimmune disorder. A total of 108 SNPs were genotyped in 384 cases of autoimmune disorder and 652 controls. The missing data is 2.82% with an average 3 SNPs per individual's genotype missing.

For those haplotype-based methods, we use two different phasing algorithms PHASE[87], GERBIL[37]. The best prediction rate achieved by linear programming for Crohn's disease and autoimmune disorder are 77.28% and 64.77%, respectively, see Table 5.2.

As shown in Table 5.2, we can find that there is a relationship between phasing methods and prediction rates. In other words, the prediction rates are depending on the phasing methods. For genotype-based algorithms, the different prediction rates is due to the different reconstruction of missing data through phasing. On the other hand, GERBIL Feasible and PHASE Feasible give a better prediction rate

| Phasing Methods (Data Set) | Population | Prediction Methods | | | | | |
|---|---|---|---|---|---|---|---|
| | | Closest Neighbor | Genotype Statistics | Haplotype Statistics | First Neighbor | Second Neighbor | Haplotype Weighting |
| GERBIL (Daly *et al.*) | Case | 54.17 | 47.22 | 52.08 | 67.39 | 67.13 | 62.19 |
| | Control | 58.85 | 64.20 | 59.67 | 57.38 | 56.96 | 84.72 |
| | **Total** | **57.11** | **57.88** | **56.85** | **61.06** | **62.44** | **76.24** |
| GERBIL Feasible (Daly *et al.*) | Case | 54.72 | 47.91 | 51.39 | 63.82 | 52.08 | 61.11 |
| | Control | 59.13 | 62.55 | 58.44 | 62.50 | 71.62 | 86.79 |
| | **Total** | **56.98** | **57.11** | **55.81** | **62.99** | **64.51** | **77.28** |
| PHASE (Daly *et al.*) | Case | 54.72 | 47.22 | 52.78 | 78.03 | 60.61 | 63.91 |
| | Control | 58.85 | 64.61 | 60.08 | 43.29 | 71.03 | 80.22 |
| | **Total** | **57.11** | **58.14** | **57.36** | **55.92** | **67.21** | **74.38** |
| PHASE Feasible (Daly *et al.*) | Case | 50.69 | 48.61 | 51.39 | 70.07 | 54.01 | 63.23 |
| | Control | 61.72 | 65.43 | 59.67 | 59.32 | 76.10 | 83.01 |
| | **Total** | **57.63** | **59.17** | **56.59** | **63.27** | **70.24** | **75.74** |
| GERBIL (Ueda *et al.*) | Case | 43.75 | 56.51 | 51.30 | 22.66 | 41.15 | 30.47 |
| | Control | 65.95 | 54.75 | 61.35 | 83.74 | 64.42 | 84.97 |
| | **Total** | **57.72** | **55.41** | **57.63** | **61.10** | **55.79** | **64.77** |

**Table 5.2.** The comparison of the prediction rates of 6 prediction methods for Crohn's Disease (Daly *et al.*)[26] and autoimmune disorder (Ueda *et al.*) [93]. Genotype data are phased by 4 methods. GERBIL [37]and PHASE [87] are statistical tools for haplotype reconstruction. For Crohn's Disease, GERBIL feasible and PHASE feasible find the respective closest feasible haplotypes of the trio data.

than GERBIL and PHASE, because GERBIL Feasible and PHASE Feasible provide a better phasing solution.

The data set of Ueda *et al.* [93] is unrelated case/control population, so the haplotype structure is much more complex than that of Daly *et al.* [26], which are described in family trios. For haplotype-based prediction methods, the complexity will affect the prediction rate. As a result, the prediction rate for Daly *et al.* is higher, as of 77.28% , while for Ueda *et al.*, is 64.77% only.

For Ueda *et al.*'s data set, we only use GERBIL [37] for phasing, because it's much faster than PHASE[87] for large population.

The best prediction results for Daly et al. [26] data are demonstrated by the Haplotype Weighting prediction algorithm when the data are phased by the GERBIL Feasible. The Figure 5.2 shows the distribution of the genotype weights for the Haplotype Weighting prediction algorithm. Since the haplotype weights are in range from -1 to 1, the resulted genotype weight equal to the sum of haplotype weights is in range from -2 to 2. The height of columns is proportional to the number of cases (positive height) and the number of controls (negative height). It is easy to see that cases prefer the right side while the controls prefer to be on the left side of the distribution.

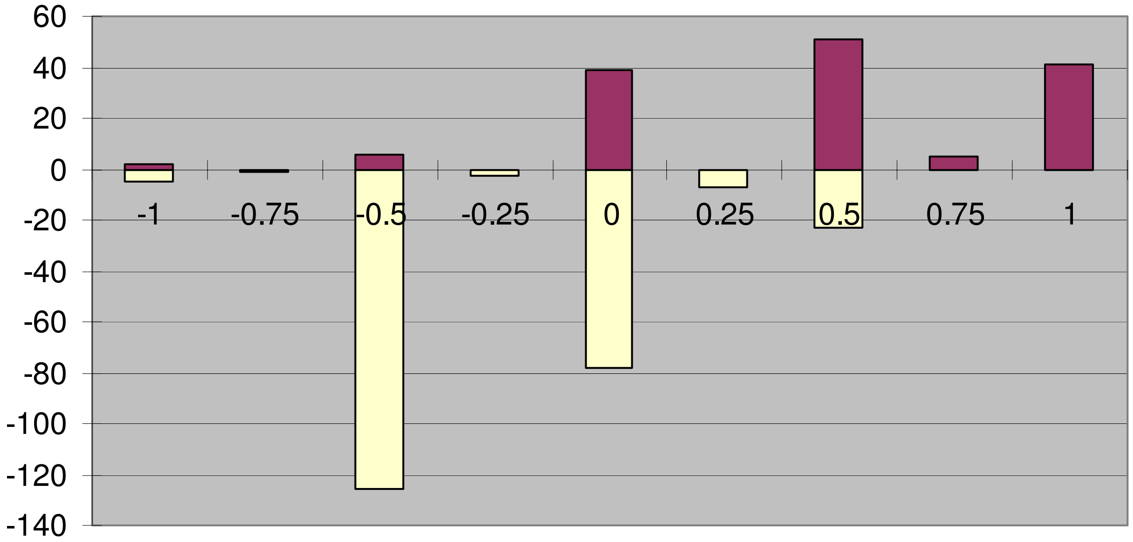


**Figure 5.2.** Distribution of the genotype weights for the Haplotype Weighting prediction algorithm. The dark columns over the median horizontal line correspond to the numbers of cases with the genotype weight in the range specified by the $x$-axis. The light columns below the median horizontal correspond to the numbers of controls within respective genotype weight range.

In Table 5.3 we compare the prediction rates of two prediction methods (Second Neighbor and Haplotype Weighting) on Daly *et al.* [26] phased by GERBIL [37] and GERBIL Feasible. We report bootstrapping rates, i. e., the 5th worst rate out of 100 runs (95% confidence) and different bootstrapping rates – averaged over 100 random samplings of 20 case and 200 control genotypes. From this table we also can find that feasible phasing allows better susceptibility prediction.

| Phasing | Population | Prediction Methods | | | |
|---|---|---|---|---|---|
| | | Second Neighbor | | Haplotype Weighting | |
| | | 95% | 200/20 | 95% | 200/20 |
| GERBIL | Case % | 67.02 | 66.89 | 58.04 | 56.65 |
| | Control % | 55.26 | 55.10 | 82.71 | 80.66 |
| | Total % | 61.13 | 58.87 | 73.58 | 71.76 |
| GERBIL Feasible | Case % | 51.32 | 50.87 | 60.84 | 59.03 |
| | Control % | 71.28 | 69.62 | 83.54 | 83.13 |
| | Total % | 64.21 | 62.83 | 75.13 | 74.16 |

**Table 5.3.** The comparison of the prediction rates of two prediction methods (Second Neighbor and Haplotype Weighting) on Daly *et al.* [26] phased by GERBIL [37] and GERBIL Feasible. We report bootstrapping rates, i. e., the 5th worst rate out of 100 runs (95% confidence) and different bootstrapping rates – averaged over 100 random choices of 20 case and 200 control genotypes.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

The most intriguing problems in genetics epidemiology are to predict genetic disease susceptibility and to associate single nucleotide polymorphisms (SNPs) with diseases. In such these studies, it is necessary to resolve the ambiguities in genetic data. The primary obstacle for ambiguity resolution is that the physical methods for separating two haplotypes from an individual genotype (phasing) are too expensive. Although computational haplotype inference is a well-explored problem, high error rates continue to deteriorate association accuracy. Secondly, it is essential to use a small subset of informative SNPs (tag SNPs) accurately representing the rest of the SNPs (tagging). Tagging can achieve budget savings by genotyping only a limited number of SNPs and computationally inferring all other SNPs. Recent successes in high throughput genotyping technologies drastically increase the length of available SNP sequences. This elevates importance of informative SNP selection for compaction of huge genetic data in order to make feasible fine genotype analysis. Finally, even if complete and accurate data is available, it is unclear if common statistical methods can determine the susceptibility of complex diseases.

This dissertation identifies the following key computational genetic epidemiology problems :

1. phasing, i.e., inferring haplotypes from available genotype data;

2. tagging, i.e., selecting informative SNPs (tags) for budget saving and data compression;

3. searching for disease-associated multi-SNP combinations and disease susceptibility prediction.

The dissertation explores above computational problems with a variety of methods, including linear algebra, graph theory, linear programming, and greedy methods. The contributions include (1)significant speed-up of popular phasing tools without compromising their quality, (2)stat-of-the-art tagging tools applied to disease association, and (3)graph-based method for disease tagging and predicting disease susceptibility.

## 6.2 Future Work

In the future, we will continue working on how to apply algorithms on bioinformatics problem. The following subsections dedicate the initial results of our future work.

### 6.2.1 Unbiased Estimates of MLR Tagging

The quality of the tag SNPs selected is dependent on the initial sample in which they are characterized. Thus it is unclear that when using standard tag SNP procedures whether the initial marker coverage is sufficiently dense to select tag SNPs, and whether the tag SNPs selected from these markers capture the required proportion of the underlying variation in the region being studied. The density of markers required will vary from one region to another depending on factors such as recombination rate, marker frequency, mutation rate and population history. If the initial marker set is too sparse the tag SNPs chosen will capture less information than a naive analysis suggests.

Weale et al. [95] proposed a "SNP-dropping" procedure for measure tag SNP performance. They assumed that the k genotyped SNPs are drawn from the same distribution as the unobserved SNPs the true performance of the tSNPs can be estimated. Each of the k SNPs is excluded in turn, the tagging procedure performed on the remaining $k - 1$ SNPs, and the proportion of the variance (R2) at the excluded SNP explained by the haplotypes formed from the tag SNPs calculated. Averaging these k values should give an unbiased estimate of the performance of the tag SNPs selected from a set of $k - 1$ SNPs. They assume both that the sample size is big enough that the haplotype frequencies are representative of the whole population, and that the observed SNPs have the same distribution as the unobserved SNPs.

Following Weale et al., we will test how our MLR tagging works in the unbiased estimation.

### 6.2.2 Protein substrate prediction

AMMP is a program designed to predict protein structures by calculating potential interactions between atoms. It is mostly used in building protein structural models according to known templates. Since AMMP is capable of calculating potential and conformational changes, it could be applied to predict favorite protein substrates. However, few studies have been performed to determine this capability. In this study, the crystal structure of caspases-3 was subjected to the binding of twenty tetra peptide substrate analogs. The binding affinities of these analogs were calculated and ranked by AMMP. The results suggest that the AMMP can predict favorite protein substrates accurately. We still don't know the result of predicting protein-protein reaction using AMMP. In the future, we will try to investigate the protein-protein interaction and improve the performance of AMMP.

**Previous Work.** Protein-protein interaction problem has been addressed in many previous and current studies. Although some bottle-necks still remain to be resolved, such as predicting main-chain conformational changes during the binding and developing efficient scoring functions; significant progress has been made in some aspects. Rigid-body algorithms have shown their promising capabilities to predict the protein-

protein docking. Kuntz.et al. [62] developed a method to explore geometrically feasible alignments of ligands and receptors of known structures. It successfully found out the receptor-ligand interface in two different systems, suggesting its possibility for generating starting conformations for energy refinement programs and interactive computer graphics routines. Later (1994-1999), Norel's group [70, 71, 72, 73, 74] developed a new method to exam the shape complementarity at the protein-protein interfaces. In their studies, receptor and ligand atoms were divided into polar and nonpolar groups and "hydrophobicity filter" was utilized. Combined with the applications of MS dot (Connolly's Molecular Surface dots) and the 3D grids, this algorithm showed a very successful result. A "soft docking" algorithm was described recently by Palma PN et al. [76]. Using their package BiGGER, the complete 6-dimensional binding spaces of both receptor and ligand will be systematically searched. Then, an interaction scoring function will be used to rank the putative docked structures. 22 out of 25 tested protein-protein complexes showed approximate binding positions to their native forms, 14 of which had very close results. In addition to these, a more recently study constructed a novel Initial-Stage Protein-Docking Algorithm (ZDOCK), where a new scoring function was developed. Majority of their tested samples got positive results during the test [17, 18, 19]. AMMP is a program designed to calculate potentials between atoms. It could be developed for protein-ligand docking. As an important function, this study will test its capability of predicting interactions between a protein and short peptides.

**Methods.** Preparation of input files of twenty tetra peptide substrate analogs. The PDB file of complex of protein caspases-3 and substrate DEVD was used as a template to build input files of other substrate analogs XEVD (X strands for any amino acids except D). Specifically, single mutations were induced on the last amino acid of the analogs by using the program O. The PDB files of twenty caspases-3/analog complexes were converted to ".ammp" file by using the program preammp. Calculating binding potentials The 3D structures of our twenty target complexes were built according to the real structure of caspases-3/DEVD. Subsequently, the binding positions of different analogs were refined by performing the energy minimization step. Finally, the binding energies were calculated by the AMMP.

**Experimental Results.** According to the calculation results, the analog DEVD showed the lowest non-bond energy, indicating that it has the highest binding affinity to the caspases-3 (top chart in Figure 6.1). EEVD was ranked the second among twenty analogs. Other analogs showed much lower binding affinities than DEVD.

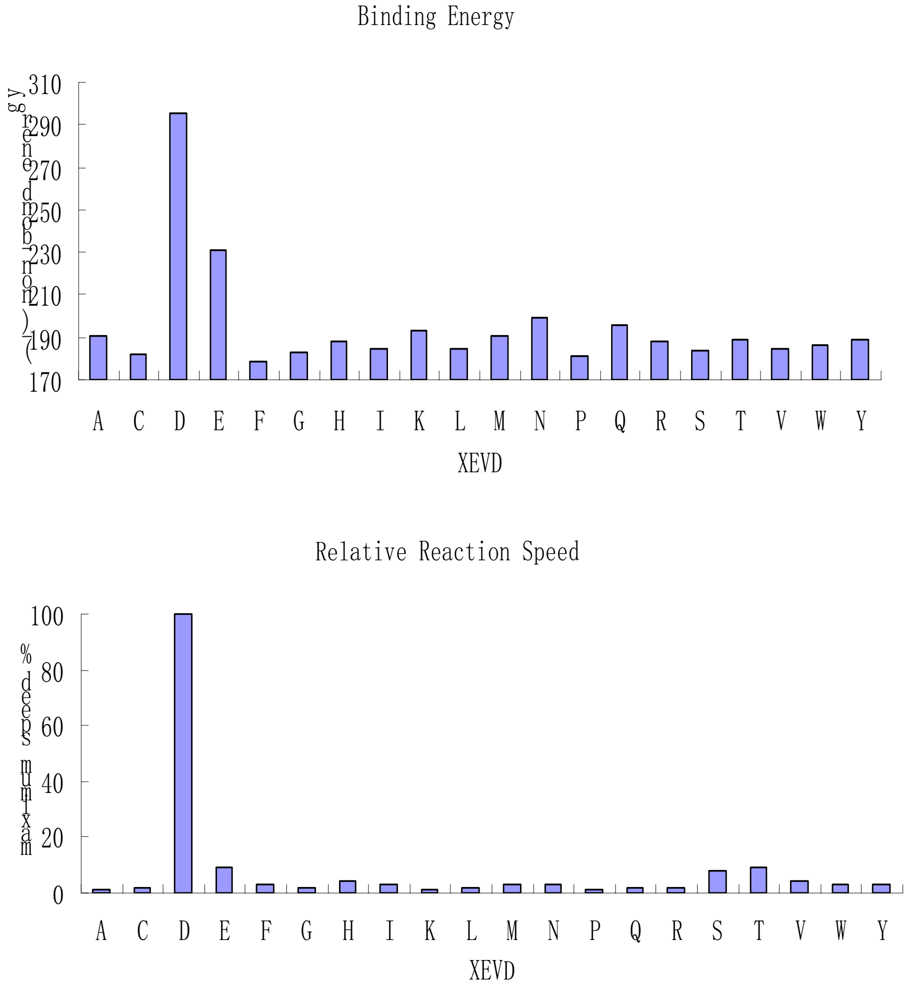


**Figure 6.1.** Prediction results from AMMP on protein binding site

### 6.2.3 Simulation of behavior of bacterial cells under specific growth conditions

A system of bacterial cells behaves in accordance with the environmental conditions prevailing at a particular instant of time. We attempts to simulate the behavior of such a system where the cells respond to the presence of an energy source. Markov chain model was used to build the quantitative description of the problem and C++ programming language was used to code the algorithm followed by the bacterial cells. The approach implemented here is general and can be applied to a variety of systems which can be represented by Markov chain models.

**Introduction.** The analysis of biological problems has recently been a very helpful tool for the biologists. The explosion of information at the genetic level has made it even more important to decipher the data presented by complex systems. The use of computer simulation for mimicking the behavior of biological systems is thus a critical area for research. Various computer models can provide insight into the working of a particular network of an organism or even the organism as a whole. The newly discovered area of systems biology is based on the mathematical modeling and simulation of underlying genetic, metabolic and signal transduction networks. Biosimulation is extensively used the pharmaceutical industry for clinical trials stage. This kind of simulation allows the elimination of potential drug candidates which can be health hazards. In the area of molecular biology, simulation can be used in conjunction with experimental work. This approach saves a lot of time for the molecular geneticist as he can know before hand the type of DNA clones are to be produced. The main object of future work is to come up with a formal model for describing the complex interaction for a species of photosynthetic bacteria and execute this model to obtain a visualization of the development of the system of this bacterial colony. This work tries to show that the behavior of a bacterial colony can be more accurately depicted if the interactions of all the components of the system are taken into consideration. Future work can include building a domain specific language for the use of simulation tools in other areas as well.

**Previous Work.** The use of simulation and modeling in the area of biological processes is not a new concept. A number of researchers have applied techniques from process algebras to this domain; a short list can be given as [51, 29, 90] Bernaschi, et al. [10] explored the use of cellular automata in the simulation of biological systems. One disadvantage of using all these techniques is that the biologist is not really well trained to understand the subtleties of these tools as their basis lie in theoretical mathematics. Hence, the visualization of this simulation is an equally important problem that has to addressed.

**Markov Method.** Any biological processes can be modeled with a set of differential equations which can be solved for the behavior of the system. These systems of equations are the quantitative description of the biological system. But this kind of description can be very different for other kinds of biological systems thus making it difficult for the model to adapt to a variety of conditions. To overcome this problem, the model presented here makes use of a set of Markov chains. Choosing Markov chains as a fundamental abstraction for this model means that this approach can be used for any system which can be reduced to a concurrent set of non-homogeneous Markov chains. Using systems of Markov chains is a general approach to modeling complicated biological systems [29]. Without loss of generality, it can be assumed that the system can converted into a system of simpler equations that can be modeled with a Markov chain. Many non-autonomous (time dependent) higher order differential equations can be represented as simple first order equations. A Markov chain is a representation of set of states of probability of moving from one state to another at a given instant. As such, Markov chains are a representation of the behavior of a non-deterministic finite state machine. For example, flipping a fair coin would have a Markov chain of order two (head/tails) and could be written as 2*2 matrix with 0.5

in each element. When the elements of the matrix are constant, the chain is referred to as a homogeneous Markov chain. The general problem of describing a system of cells requires that the coefficients of the chain change with time and cell state so that non-homogeneous chains will be used. Psuedocode Bacteria actively seek out sources of food, energy, oxygen and general chemoattractants by swimming towards them. The sequence of steps presented below assumes the knowledge of the movement of bacteria in response to various environmental cues.

While the entire algorithm used in the original work is beyond the scope of this project, a general pseudocode is given below:

1. At time t =0 a vector of capacity 1000 elements is created.

2. The length of the vector is set to 100 elements which represent the bugs at t=0.

3. A light source is providing as energy in a particular area by using the appropriate tools in the POV-Ray software.

4. Based on the probability defined by the probability transition matrix, the next state of each of the cells is determined. For simplicity, the total time is taken to be 1000 units.

5. For each time step, the probablitiy functions of the cells are re-evaluated and the next step is obtained based on previous states.

6. Finally, a pov (Persistence of vision) file is generated by the program for each of the time step from 0 to 999.

7. These image files can be run as animation to show the visual effects of the modeling. It should be noted here that the core work for this simulation program was already done [51]. This project has modified certain energy conditions for the bacterial growth hence all the header files and supporting software was used as before (with permission of the corresponding authors).

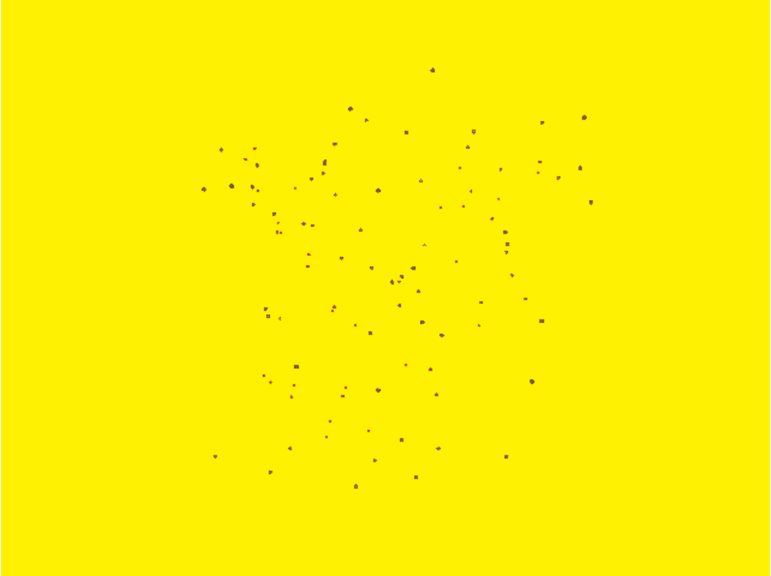


**Figure 6.2.** Bacterial simulation at time t = 0

**Experimental Results.** One of the major challenges in modeling bacterial chemotaxis is understanding how a very small organism can sense a gradient of a chemical light. For example, photosynthetic bacteria swim into bright areas, but are roughly the same size as the wavelength of light and so cannot sense local differences in brightness directly. The bacteria apply a random walk algorithm to find optimal conditions. The photosynthetic bacteria Rhodobacter Sphaeroides was chosen as a model system. R. Sphaeroides will actively swim towards regions of greater light using their flagellar motors to alternatively swim and tumble. R. Sphaeroides is modeled here as one Markov chain representing the actions individuals may choose such as tumble, laze, adjust speed, grow and divide. These choices are the function of the individuals environment and history. At time t =0, the state can be shown as Figure 6.2,

These choices are functions of the individual's environment and history. The probability of dividing, for example, is a function of the size of the bacterium; individuals that were in optimal conditions grew faster and thus had more progeny. When applied to our present system the result after time t=999 can be seen as Figure 6.3

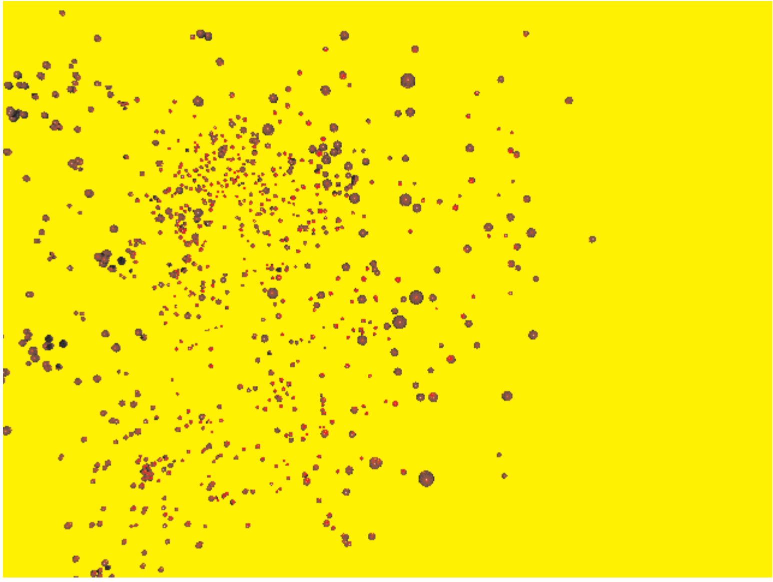


**Figure 6.3.** Bacterial simulation at time t = 999

The clearly seen bacterial growth is a function of the initial arrangement of the bacteria and the growth due to the pre-existing probabilities at that instant of time. This initial work shows that a bacterial colony can be adequately described not only by the governing mathematical equations but also by the Markov chains representations of these systems. The formation of Markov chains gives the flexibility to use the results of this work in any other kind of system biological or not. The system modeled can also be animated using animation software and thus can be used as visual aid for the practicing biologist.

# BIBLIOGRAPHY

[1] Affymetrix (2005) *http://www.affymetrix.com/products/arrays/.*

[2] Basic Classification Trees, *http://www.ece.wisc.edu/ nowak/ece901/lecture11.pdf*

[3] ILOG CPLEX, *http://www.ilog.com*

[4] LPsolve, *http://www.netlib.org*

[5] International HapMap Consortium (2003) The International HapMap Project. *Nature*, **426**, 789–796, *http://www.hapmap.org.*

[6] Ackerman, H., Usen, S., Mott, R., Richardson, A., Sisay-Joof, F., Katundu, P., Taylor, T., Ward, R., Molyneux, M., Pinder, M., Kwiatkowski, D.P. (2003) 'Haplotypic analysis of the TNF locus by association efficiency and entropy', *Genome Biology*, Vol.4, pp. 24

[7] Anderson, M. (2001) 'Crohn's: An Autoimmune or Bacteria-Related Disease?', *The Scientist*, 22:15-16.

[8] Avi-Itzhak, H.I., Su, X. and de la Vega, F.M. (2003) 'Selection of minimum subsets of single nucleotide polymorphism to capture haplotype block diversity', *Proceedings of Pacific Symposium on Biocomputing*, Vol. 8, pp. 466–477.

[9] Bafna, V., Halldorsson, B.V., Schwartz, R.S., Clark, A.G. and Istrail, S. (2003) 'Haplotypes and informative SNP selection algorithms: don't block out information', *Proceedings of the Seventh International Conference on Research in Computational Molecular Biology*, pp. 19–27.

[10] Bernaschi, M. and Castoglione, F. “Design and implementation of an immune system simulator,” Computers in Biology and Medicine, 2001, vol .31, pp. 303-331.

[11] Botstein, D., Risch, N. (2003) ‘Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease’, *Nature Genetics*, 33:228-237.

[12] Brinza, D. and Zelikovsky, A. (2006) 2SNP: Scalable Phasing Based on 2-SNP Haplotypes, *Bioinformatics*, **22(3)**, 371–373.

[13] Brinza, D., He, J., Mao, W. and Zelikovsky, A. (2005). ‘Family Trio Phasing and Missing Data Recovery’, *International Journal on Bioinformatics Research and Applications, 1(2)*, pp. 221–229.

[14] Brinza, D., He, J., Mao, W. and Zelikovsky, A. (2005) “Phasing and Missing data recovery in Family Trios,” Proceedings of ICCS 2005 LNCS 3515, June 2005, 1011-1019.

[15] Brown, D.G. and Harrower, I.M. (2004) ‘A new integer programming formulation for the pure parsimony problem in the haplotype association’, *Workshop on Algorithms in Bioinformatics*, v.3240.

[16] Carlson, C.S., Eberle, M.A., Rieder, M.J., Yi, Q., Kruglyak, L. and Nickerson, D.A. (2004) ‘Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium’, *American Journal of Human Genetics*, Vol. 74, No. 1, pp. 106–120.

[17] Chen R, Li L, Weng Z. ZDOCK: an initial-stage protein-docking algorithm. Proteins. 2003 Jul 1;52(1):80-7.

[18] Chen R, Tong W, Mintseris J, Li L, Weng Z. ZDOCK predictions for the CAPRI challenge. Proteins. 2003 Jul 1;52(1):68-73.

[19] Chen R, Weng Z. A novel shape complementarity scoring function for protein-protein docking. Proteins. 2003 May 15;51(3):397-408.

[20] Clark, A. . Inference of haplotypes from PCR-amplified samples of diploid populations. Mol. Biol. Evol, 7:111–122, 1990.

[21] Clark, A., Weiss, K., Nickerson, D., Taylor, S., Buchanan, A., Stengard, J., Salomaa, V., Vartiainen, E., Perola, M., Boerwinkle, E., et al. (1998) 'Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase', *American Journal of Human Genetics*, Vol. 63, pp. 595–612.

[22] Clark, A. (2003) 'Finding genes underlying risk of complex disease by linkage disequilibrium mapping', *Current Opinion in Genetics & Development*, Vol. 13, No. 3, pp. 296–302.

[23] Chapman, J.M., Cooper, J.D., Todd, J.A. and Clayton, D.G. (2003). 'Detecting disease associations due to linkage disequilibrium using haplotype tags: a class of tests and the determinants of statistical power', Human Heredity, Vol. 56, pp. 18–31.

[24] Chung, R.H. and Gusfield, D. Perfect phylogeny haplotyper: Haplotype inferral using a tree model. Bioinformatics, 19(6):780–781, 2003.

[25] Chung, R.H. and Gusfield, D. Empirical exploration of perfect phylogeny haplotyping and haplotypers. In Proceedings of COCOON 03 The 9'th International Conference on Computing and Combinatorics, volume 2697 of LNCS, pages 5–19, 2003.

[26] Daly, M., Rioux, J., Schaffner, S., Hudson, T. and Lander, E. (2001) ‘High resolution haplotype structure in the human genome’, *Nature Genetics*, Vol. 29, pp. 229–232.

[27] Efron, B. (1983), ”Estimating the error rate of a prediction rule: Improvement on cross-validation,” *J. of the American Statistical Association*, 78, 316-331.

[28] Efron, B. and Tibshirani, R.J. (1993), *An Introduction to the Bootstrap*, London: Chapman & Hall.

[29] Ermentrout, B Computational modeling of Genetic Biochemical networks, MIT press, 2001

[30] Eskin, E., Halperin, E. and Karp, R. (2003) ‘Efficient reconstruction of haplotype structure via perfect phylogeny’, *Journal of Bioinformatics and Computational Biology*, Vol. 1, No. 1, pp. 1–20.

[31] Falk, C. T. and Rubinstein, P. (1987) ‘Haplotype relative risks: an easy reliable way to construct a proper control sample for risk calculations’, *Ann Hum Genet.* 1987 Jul;51 (Pt 3):227-33.

[32] Gabriel, G., Schaffner, S., Nguyen, H., Moore, J., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E., Daly, M. and Altshuler, D. (2002) ‘The structure of haplotype blocks in the human genome’, *Science*, Vol. 296, pp. 2225–2229.

[33] Goldestein, D. and Weale, M. (2001) ‘Population genomics:Linkage disequilibrium holds the key’, *Current Biology*, 11:576-579.

[34] Gusfield, D.. Inference of haplotypes from samples of diploid populations: complexity and algorithms. Journal of computational biology, 8(3), 2001.

[35] Gusfield, D. (2003) ‘Haplotype inference by pure parsimony’, *In R. Baeza-Yates, E. Chavez, and M. Chrochemore,ed. 14’th Annual Symposium on Combinatorial Pattern Matching*, v. 2676 of Springer LNCS, 144–155.

[36] Forton, J., Kwiatkowshi, D., Rockett, K., Luoni, G., Kimber, M. and Hull, J. (2005) ‘Accuracy of Haplotype Reconstruction from Haplotype-Tagging Single-Nucleotide Polymorphisms’, *American Journal of Human Genetics* Vol. 76, pp 438–448.

[37] Kimmel, G., and Shamir R.(2004). ‘GERBIL: Genotype resolution and block identification using likelihood’, *PNAS*, Vol. 102, pp 158–162.

[38] Halperin, E. and Eskin, E. Haplotype reconstruction from genotype data using imperfect phylogeny. Bioinformatics. Advance Access published on February 26, 2004.

[39] Halperin, E. and Karp, R. M. On the greedy set cover algorithm. In preperation, 2003.

[40] Halperin, E. and Karp, R. M. Perfect phylogeny and haplotype assignment. RECOMB, 2004.

[41] Halperin, E., Kimmel, G. and Shamir, R. (2005) ‘Tag SNP Selection in Genotype Data for Maximizing SNP Prediciton Accuracy’, *Bioinformatics*, Vol. 21, pp. 195-203.

[42] Halldorsson, B.V., Bafna, V., Lippert, R., Schwartz, R., de la Vega, F.M., Clark, A.G. and Istrail, S. (2004) ‘Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies’, *Genome Research* Vol. 14, pp. 1633–1640.

[43] He, J. and Zelikovsky, A. (2004) ‘Linear Reduction Methods for Tag SNP Selection’, *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology (EMBC’04)*, pp. 2840–2843.

[44] He, J. and Zelikovsky, A. (2004) ‘Linear Reduction for Haplotype Inference’, , *Proceedings of the Workshop on Algorithms in Bioinformatics (WABI’04)*, Vol. 3240, pp. 242–253.

[45] He, J. and Zelikovsky, A. (2005) ‘Linear Reduction Method for Predictive and Informative Tag SNP Selection’, , *International Journal Bioinformtics Research and Applications*, Vol 3, pp. 249–260.

[46] He, J. and Zelikovsky, A. (2006) “Tag SNP Selection Based on multiple Linear Regression,” Proc. of Intl Conf on Computational Science (ICCS 2006), May 2006, LNCS 3992, pp. 750-757

[47] He, J. and Zelikovsky, A. (2006) “Haplotype Tagging based on SVM SNP Prediction,” Proc. IEEE Intl Conf on Granular Computing (GRC 2006), May 2006, pp. 758-761

[48] He, J. and Zelikovsky, A. (2006) “MLS-tagging: Genotype Tagging Based on Multivariate Linear Regression,” Application notes, Bioinformatics, to appear

[49] He, J. and Zelikovsky, A. (2006) “Multiple Linear Regression for Index SNP Selection on Unphased Genotypes,” Proc. International Conf. of the IEEE Engineering in Medicine and Biology (EMBC’06), September 2006, to appear.

[50] Brinza, D., He, J. and Zelikovsky, A. “Combinatorial Search Methods for Multi-SNP Disease Association,” Proc. International Conf. of the IEEE Engineering in Medicine and Biology (EMBC’06), September 2006, to appear.

[51] Harrison, W. and Harrison R., "Domain specific languages for cellular interactions," Proceedings of the 26th annual IEEE conference on engineering in Medicine and Biology, 2004, pp 3019-3022.

[52] Herbert, A., Gerry, N.P., McQueen, M.B. (2006) A Common Genetic Variant Is Associated with Adult and Childhood Obesity, SCIENCE, 312, 279–284.

[53] Hjorth, J.S.U. (1994), *Computer Intensive Statistical Methods Validation*, Model Selection, and Bootstrap, London: Chapman & Hall.

[54] Hudson, R. (1990) 'Gene genealogies and the coalescent process', *Oxford Survey of Evolutionary Biology*, Vol. 7, pp. 1–44.

[55] Obtaining Unbiased Estimates of Tagging SNP Performance, (2005) Annals of Human Genetics vol. 69, page 1–8.

[56] Judson, R., Salisbury, B., Schneider, J., Windemuth, A. and Stephens, J.C. (2002) 'How many SNPs does a genome-wide haplotype map require?', *Pharmacogenomics*, Vol. 3, pp. 379–391.

[57] Ke, X. and Cardon, LR. (2003) 'Efficient selective screening of haplotype tag SNPs', *Bioinformatics*, Vol. 170, pp. 287-288.

[58] Kam, N., Cohen, I. and Harel, D.,"Modeling biological reactivity: statecharts vs. Boolean logic," Second International conference on Systems Biology, 2001.

[59] Knapp, M., Seuchter, S.A. and Baur, M.P. (1987) 'The haplotype-relative-risk (HRR) method for analysis of association in nuclear families', *Am J Hum Genet.* 1993 Jun;52(6):1085-93.

[60] Kimmel, G. and Shamir R.. (2005) A Block-Free Hidden Markov Model for Genotypes and Its Application to Disease Association. *J. of Computational Biology*, Vol. 12, No. 10: 1243-1260.

[61] Kim Gene and Kim, MyungHo. Application of Support Vector Machine to detect an association between a disease or trait and multiple SNP variations, (http://xxx.lanl.gov/abs/cs.CC/0104015)

[62] Kuntz I, Blaney J, Oatley S, Langridge R, Ferrin T. A geometric approach to macromolecule-ligand interactions. J Mol Biol 1982; 161: 269-288.

[63] Li, J. and Jiang, J. (2003) 'Efficient Rule-Based Haplotyping Algorithm for Pedigree Data. In Proc.', *International Conference on Research in Computational Molecular Biology*, 197-206

[64] Lin, S., Chakravarti, A. and Cutler, D.J. (2004) 'Haplotype and Missing Data Inference in Nuclear Families', *Genome Res*,14(8):1624-32.

[65] Lin, S., Cutler, D., Zwick, M. and Cahkravarti, A. (2003) 'Haplotype inference in random population samples'. *America Journal of Human Genetics*, Vol 71, pp 1129–1137.

[66] Listgarten, J., Damaraju, S., Poulin B., Cook, L., Dufour, J., Driga, A., Mackey, J., Wishart, D., Greiner,R., and Zanke, B.. (2004) Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms. *Clinical Cancer Research*, Vol. 10, 2725C2737, 2004.

[67] Marchini, J., Donnelley , P. and Cardon, L.R, (2005) Genome-wide strategies for detecting multiple loci that influence complex diseases, *Nature Genetics*, **37**, 413–417.

[68] Merikangas, KR., Risch, N. (2003) 'Will the genomics revolution revolutionize psychiatry', *The American Journal of Psychiatry*, 160:625-635.

[69] Niu, L., Qin, Z., Xu, X. and Liu, J.S.. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. Am. J. Hum. Genet, 70:157–169, 2002.

[70] Norel R, Lin SL, Wolfson H, Nussinov R. Shape complementarity at protein-protein inter-faces. Biopolymers 1994; 34: 933-940.

[71] Norel R, Petrey D, Wolfson H, Nussinov R. Examination of shape complementarity in docking of unbound proteins. Proteins 1999; 35: 403-419.

[72] Norel R, Wolfson H, Nussinov R. Small molecule recognition: solid angles surface representation and shape complementarity. Comb Chem High Throughput Screen 1999; 2: 177-191.

[73] Norel R, Fischer D, Wolfson H, Nussinov R. Molecular surface recognition by a computer vision based technique. Prot Eng 1994; 7: 39-46.

[74] Norel R, Lin SL, Wolfson H, Nussinov R. Molecular surface complementarity at protein-protein interfaces: the critical role played by surface normals at well placed, sparse points in docking. J Mol Biol 1995; 252: 263-273.

[75] Orzack, S., Gusfield, D. and Stanton, V.. The absolute and relative accuracy of haplotype inferral methods and a consensus approach to haplotype inferral. Abstract Nr 115 in Am. Society of Human Genetics, Supplement 2001.

[76] Palma PN, Krippahl L, Wampler JE, Moura JJG. BIGGER: a new soft docking algorithm for predicting protein interactions. Proteins 2000; 39: 178-194.

[77] Pasaniuc, B. and Mandoiu, I. Highly Scalable Genotype Phasing by Entropy Minimization, submitted to EMBC06.

[78] Patil, N., Berno, A., Hinds, D., Barrett, W., Doshi, J., Hacker, C., Kautzer, C., Lee, D., Marjoribanks, C., McDonough, D., Nguyen, B., Norris, M., Sheehan, J., Shen, N., Stern, D., Stokowski, R., Thomas, D., Trulson, M., Vyas, K., Frazer, K., Fodor, S. and Cox, D. (2001) 'Blocks of limited haplotype diversity revealed

by high-resolution scanning of human chromosome', *Science*, Vol. 294, pp. 1719–1723.

[79] Phase Phamily analysis notes.

[80] Plutowski, M., Sakata, S., and White, H. (1994), "Cross-validation estimates IMSE," in Cowan, J.D., Tesauro, G., and Alspector, J. (eds.) *Advances in Neural Information Processing Systems* 6, San Mateo, CA: Morgan Kaufman, pp. 391-398.

[81] Qin, Z., Niu, T., and Liu, J. (2002) 'Partitioning-Ligation-Expectation- Maximization algorithm for haplotype inference with single-nucleotide polymorphisms', *American Journal of Human Genetics*, Vol. 71, pp. 1242–1247.

[82] Regev, A., Silverman, W., and Shapiro, E. "Representation and simulation of biochemical processes using the calculus process algebra," Pacific Symposium on Biocomputing, 2001, vol. 6, pp. 459-470.

[83] Sebastiani, P., Lazarus, R., Weiss, S., Kunkel, L., Kohane, I., and Ramoni, M. (2003) 'Minimal haplotype tagging', *Proceedings of the National Academy of Sciences*, Vol. 100, pp. 9900–9905.

[84] Serretti, A. and Smeraldi, E. (2004) 'Neural network analysis in pharmacogenetics of mood disorders', *BMC Medical Genetics*, 5:27.

[85] Shao, J. and Tu, D. (1995), *The Jackknife and Bootstrap*, New York: Springer-Verlag.

[86] Spinola, M., Meyer, P., Kammerer, S. et al. (2006) Association of the PDCD5 Locus With Lung Cancer Risk and Prognosis in Smokers, American Journal of Clinical Oncology, 24:11.

[87] Stephens, M., Smith, N., and Donnelly, P.. (2001) 'A new statistical method for haplotype reconstruction from population data', it American Journal of Human Genetics, Vol. 68, pp. 978–989.

[88] Stram, D., Haiman, C., Hirschhorn, J., Altshuler, D., Kolonel, L., Henderson, B. and Pike, M. (2003). 'Choosing haplotype-tagging SNPs based on unphased genotype data using as preliminary sample of unrelated subjects with an example from the multiethnic cohort study', *Human Heredity*, Vol. 55, pp. 27–36.

[89] Tang, Y.C., Jin, B. and Zhang, Y.Q., "Granular Support Vector Machines with Association Rules Mining for Protein Homology Prediction," *Special Issue on Computational Intelligence Techniques in Bioinformatics, Artificial Intelligence in Medicine*, vol. 35, no. 2, pp. 121–134, 2005.

[90] Taylor, B. and Zhulin, I. "In search of higher energy: metabolism -dependent behavior in bacteria," Molecular Microbiology, vol. 28, pp. 683-690, 1998.

[91] Thornberry, N. A., Rano, T. A., Peterson, E. P., Rasper, D. M., Timkey, T., Garcia-Calvo, M., Houtzager, V. M., Nordstrom, P. A., Roy, S., Vaillancourt, J. P., Chapman, K. T. and Nicholson, D. W. (1997). A combinatorial approach defines specificities of members of the caspase family and granzyme B. Functional relationships established for key mediators of apoptosis. J Biol Chem 272, 17907-11.

[92] Tomita, Y., Yokota, M. and Honda, H. (2005) Classification method for prediction of multifactorial disease development using interaction between genetic and environmental factors, *IEEE computational systems bioinformatics conference*, abstract.

[93] Ueda, H., Howson, J.M.M., Esposito, L. et al. (2003) Association of the T Cell Regulatory Gene CTLA4 with Susceptibility to Autoimmune Disease, *Nature*, **423**, 506–511.

[94] Vapnik, V. and Cortes, C. , “Support Vector Networks”,*Machine Learning*, vol. 20, pp. 273–293, 1995.

[95] Weale ME, Depondt C, Macdonald SJ, Smith A, Lai PS, Shorvon SD, Wood NW, Goldstein DB (2003) Selection and evaluation of tagging SNPs in the neuronal-sodium-channel gene SCN1A: implications for linkage-disequilibrium gene mapping. Am J Hum Genet 73:551C565

[96] Weiss, S.M. and Kulikowski, C.A. (1991), *Computer Systems That Learn*, Morgan Kaufmann.

[97] Weidong Mao, Jingwu He, Dumitru Brinza and Alex Zelikovsky, ”A Combinatorial Method for Predicting Genetic Susceptibility to Complex Diseases,” Proceedings of International Conference of the IEEE Engineering in Medicine and Biology (EMBC’05), September 2005, 1832-1835.

[98] Zhang, K., Qin, Z., Liu, J., Chen, T., Waterman, M., and Sun, F. (2004) ‘Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies’, *Genome Research*, Vol. 14, pp. 908–916.

[99] Zhao, H., Pfiffer, R. and Gail, MH. (2003) ‘Haplotype analysis in population genetics and association studies’, *Phamacogenomics*, 4:171-178.