# A CO-OPERATIVE CO-EVOLUTIONARY GENETIC ALGORITHM FOR HAPLOTYPE PATTERN DETECTION IN CASE-CONTROL DATA

MOHAMMED UDDIN

# A Co-operative Co-evolutionary Genetic Algorithm for Haplotype Pattern Detection in Case-Control Data

by

© Mohammed Uddin

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of Science

Department of Computer Science

Memorial University of Newfoundland

January 23, 2009

St. John's Newfoundland

## Abstract

Genomic variations such as Single Nucleotide Polymorphisms (SNP) and their underlying haplotype patterns in case-control cohorts are used to identify genes associated with diseases. Complex diseases involve multiple genes which may be distributed over the genome. A popular technique for detecting such markers and patterns is the sliding window technique using statistical models. However, the statistical techniques used are computationally expensive, and derived patterns are typically restricted both in length and to consist of contiguous markers. In this thesis, we have developed a cooperative coevolutionary genetic algorithm (CCGA) that can compute both contiguous and non-contiguous marker haplotype patterns from case-control haplotype data; moreover, this algorithm can tolerate missing/ambiguous positions in haplotype data arising during haplotype phasing from genotypes.

We have tested our algorithm on three case-control cohorts (the Ankylosing Spondilitis (AS) inflammatory arthritis cohorts from Alberta (AL) and Newfoundland (NF) populations (genotyped for the IL1 gene cluster on chromosome 2) and the Japanese Schizophrenia cohort (genotyped for the Netrin G1 gene on chromosome 1)). The results obtained using our CCGA are in strong accordance with previously published results. Specifically, (1) in the AL spondylitis cohort, we have found significant haplotype patterns ($p < 0.0005$ and haplotype risk ratio $\geq 1.5$) that confer susceptibility of four genes (IL1A, IL1B, IL1F7 and IL1F10) with AS, three of which (IL1A, IL1B, IL1F10) were confirmed by two independent studies; and (2) in the Japanese schizophrenia cohort, 7 SNPs (rs4481881, rs4307594, rs3924253, rs4132604, rs1373336, rs1444042, and rs96501) and their haplotypes showed significant ($p < 0.0005$ and hap-

lotype risk ratio $\geq 1.5$) association with schizophrenia, the most significant of which (rs4307594, rs3924253, and rs1373336) were confirmed by two independent studies.

## Acknowledgements

I would like to express my gratitude to all those who helped to complete this thesis. I am thankful to Dr. Tina Yu for her guidance in the algorithm design and testing. I am grateful to Dr. Todd Wareham for guiding me throughout my graduate studies. I am deeply grateful to Dr. Proton Rahman for introducing me to the world of genetics. I am thankful to Dr. Takeo Yoshikawa for his cooperation with the Schizophrenia dataset and Dr. Michael Nothnagel and Nicole Roslin for their help with statistics. I am also greatly indebted to the computer science general office staff, system administrators and my colleagues at the population therapeutic research group (PRTG).

Finally, I would like to give my special thanks to my parents and wife; their unconditional support made it possible for me to complete this thesis.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

The sequencing of the complete human genome gives us the hope to isolate or detect the genomic regions or genes that are responsible for various genetic diseases. Human genetic diseases are separated into two basic classes - Mendelian diseases (characterized by mutation in a single gene) and complex or multi-factorial diseases (characterized by multiple mutations distributed across multiple genes) [8, 42, 50]. The relationship between the genotype and its associated complex disease phenotype is still an open challenge. There are several bottlenecks that hinder the process of investigating the biological activity of complex disease at the molecular level [19]. In particular, the availability of large genetic datasets for complex disease analysis requires a computationally feasible approach, and many proposed algorithms for investigating problems related to complex diseases are simply too expensive to use.

Investigations of complex diseases mainly have focused on the analysis of hu-

man deoxyribonucleic acid (DNA) variations known as single nucleotide polymorphisms (SNP). The common approach applies classical statistical methods that are computationally intensive and restricted to computing patterns from a small number of contiguous SNPs [14, 15]. This restriction to contiguous SNPs means that this approach is unable to detect multiple widely distributed genes that are characteristic of complex diseases [49]. Different approaches have been proposed to alleviate these problems; unfortunately these approaches frequently neglect biologically relevant information such as human genome variation. We need a biologically relevant and computationally useful approach that can solve the problem of detecting genomic regions for complex diseases.

## 1.2 Objectives

The intent of this thesis is to contribute to the area of investigating complex disease by proposing a computationally feasible algorithm for the detection of multiple mutations in the human genome associated with complex diseases that incorporates knowledge of human genetic variation. Detecting susceptible regions over the entire human genome is not the focus of this thesis; instead, our proposed algorithm attempts to alleviate the restrictions that the common statistical method faces while computing susceptibility in a segment of human genome.

## 1.3 Contribution

The main contribution of the thesis is to propose a practical algorithm for haplotype pattern detection using case-control SNP data. The primary consideration while designing this algorithm is to overcome the problems that previous statistical models could not handle. The specific contributions are as follows:

1. We have designed a cooperative coevolutionary genetic algorithm (CCGA) to detect disease susceptible SNPs and their underlying haplotypes in a segment of any human chromosome using case-control data (Section 4.3). There is no existing CCGA scheme that solves the problem. The proposed algorithm computes susceptibility of contiguous and non contiguous haplotypes which allows detection of disease susceptible genes that are physically far apart from each other in the genome.

2. The current technology produces SNP data which are also known as genotypes. There are different types of algorithms that construct haplotype data from these genotypes. After the construction of haplotypes, there can exist missing or ambiguous data. We have proposed a new algorithm that can handle ambiguous or missing data while computing susceptibility of haplotypes (Section 4.3.3).

3. We have applied our CCGA algorithm to three published datasets (Section 5.1). Previous analyses of these datasets used statistical techniques to detect susceptible SNPs and haplotypes. Results obtained by our CCGA are in strong accordance with previously published analyses. Moreover, the computational effort required by our approach is much less than that of the statistical methods.

Preliminary descriptions of these contributions have appeared in [61, 62, 63].

## 1.4 Organization

In Chapter 2, we review background knowledge necessary for the problem examined in this thesis. In Section 2.1, we review classical genetics and its molecular implementations. This includes an extended description of a particularly important kind of molecular sequence variation called SNPs. In Section 2.2, we review the analytical models of these variations for both Mendelian and complex diseases.

In Chapter 3, we formulate the problem of detecting haplotype patterns for a complex disease in a case-control cohort. In Section 3.1, we review the general problem of detecting genomic regions for diseases and formulate our specific problem. In Section 3.2, we also review previously proposed methods that have been successful in the detection of susceptible SNPs and haplotypes in complex diseases. The advantages and disadvantages of each method are also discussed, giving a set of requirements for a desired approach that can handle the problems pointed out for the previous approaches.

In Chapter 4, we propose an algorithm that satisfies the requirements listed in Chapter 3. In Section 4.1, we outline the basic mechanism of a standard genetic algorithm. In Section 4.2, we then describe the basic mechanisms of a CCGA scheme as well as the differences between CCGA and standard genetic algorithms. Finally, in Section 4.3, we give the details of our proposed CCGA scheme for the detection of susceptible SNPs and their underlying haplotypes.

In Chapter 5, we analyze the performance of our CCGA. In Sections 5.1 and 5.2,

the datasets and important parameters for the experiments are discussed. In Section 5.3, this performance is analyzed by assessing the evolutionary force of the algorithm with statistical significance tests. In Section 5.4, the quality of results obtained by the proposed CCGA is compared with published results. In Section 5.5, the chapter concludes with a discussion of the advantages and the disadvantages of the proposed algorithm.

Finally, in Chapter 6, we give our conclusions and sketch a road map of directions for future research.

# Chapter 2

# Background

Human genetics investigates inheritance both in the classical genetic sense and at the molecular level. Current advances in human genetics are mainly focused on human disease investigation. Molecular mechanisms are key to understanding human disease; however, it is also critical to adapt classical genetics models to work at the molecular level.

In this chapter, we review basic genetics and its relationship to the analysis of disease. In Section 2.1, we introduce the basic entities of classical human genetics and their associated molecular mechanisms. Models for analyzing human genetic disease are discussed in Section 2.2.

## 2.1 Human Genetics

Human genetics gives us knowledge of inheritance of characteristics that occur in human beings. The first achievements in genetics were in the 19th century, when Gregor Mendel investigated plant hybridization and established the basic theory of

inheritance. In modern genetics, the various concepts in the classical theory of inheritance can be understood at the molecular level. In particular, molecular genetic variation gives important insights about the molecular basis of complex diseases.

In Sections 2.1.1 and 2.1.2, we outline the classical genetic concepts and their implementations in molecular genetics (see Figure 2.1). In Section 2.1.3, the properties and definitions related to a particularly important molecular variation, namely SNPs, are discussed (see Figure 2.2). More details may be found in standard textbooks such as [56, 51].

### 2.1.1 Classical Genetics

In this section, we will review various concepts from classical genetics (see Figure 2.1(A)).

#### 2.1.1.1 Gene

In classical genetics, a gene corresponds to a particular characteristic of an organism. An allele corresponds to a particular state of that characteristic. For example, Gregor Mendel experimented with the color and texture characteristics of pea plants. He hybridized smooth yellow peas with wrinkly green peas and the offspring produced peas with yellow color and smooth skin. Such experiments showed that the offspring plant inherited the color and texture characteristics from its parents' plants. More complex characteristics may actually be encoded by a group of genes (i.e. genes for human eye color).

#### 2.1.1.2 Chromosome and Locus

A chromosome is a unit of heredity containing a linearly ordered sequence of genes. The concept of chromosome was introduced by Karl Wilhelm von Ngeli in 1842 when he was investigating plant cells. In 1910, Thomas Hunt Morgan showed that chromosomes are the carriers of genes. Different organisms are characterized by the number of distinct chromosomes and the number of copies of each chromosome that they have. In human beings, there are 23 pairs of chromosomes where each member of a pair is inherited from one of the parents. Thus, each gene on a human chromosome has two alleles corresponding to the alleles inherited from each parent. If the alleles are the same for both parents, the gene is homozygous, and if the alleles are different, it is heterozygous. The set of alleles for a parent is known as a haplotype, and the set of allele-pairs from the two parents is known as a genotype (see Figure 2.2).

In genetics, the term locus is commonly used to refer to genetic functional regions. The chromosomal position of a gene is also known as a locus [56]. Figure 2.1($A$) shows a locus on a chromosome corresponding to a gene.

### 2.1.2 Human Molecular Genetics

In this section, we will discuss some of the basic terminology of molecular genetics (see Figure 2.1(B)). More detailed descriptions are given in [56].

#### 2.1.2.1 Protein

The first molecule that was believed to be the basic element of any biological function was protein. Some proteins called enzymes catalyze biochemical reactions which are

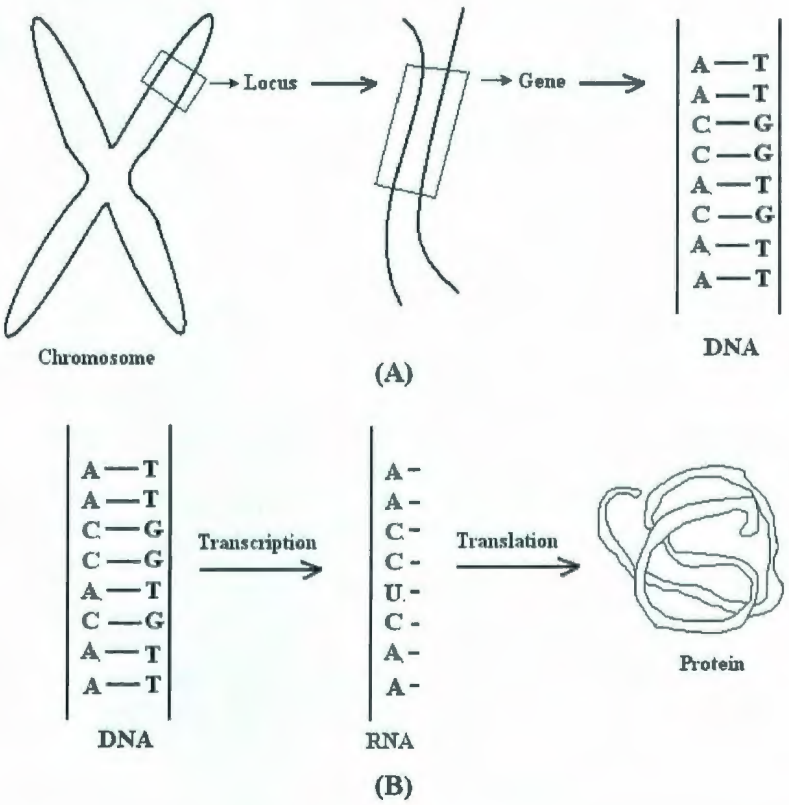


Fig. 2.1: Concepts in Classical and Molecular Genetics. (A) Classical genetic concepts and their implementations in DNA. (B) The relationship between DNA, RNA and protein. The relationships shown in (B) form what is known as the Central Dogma of molecular genetics [56].

crucial to metabolism [57]. The role of protein was first described by James Sumner who showed that the enzyme urease is a protein. Proteins are essential parts of organisms and participate in every process within cells. Proteins also have structural or mechanical functions: i.e. actin and myosin proteins maintain cell shape.

The first protein sequence was not available until 1958 when Frederick Sanger sequenced the insulin protein. Proteins are strings of amino acids; 20 primary amino acids are known to exist. The function of a protein is dependent on its sequence,

specifically on the 3 dimensional folded shape of the sequence.

#### 2.1.2.2 Deoxyribonucleic Acid (DNA) and Ribonucleic Acid (RNA)

Until the discovery of DNA, scientists believed protein alone was responsible for the functions of human cells, including inheritance. DNA consists of an array of genes which encode the proteins that function in the human body. The construction of a protein from a DNA sequence is a two fold task. Before explaining protein formation, we need to know the basic structure of DNA.

In this thesis, we will focus on the DNA that makes up the 23 pairs of chromosomes inside the nuclei of human cells.[1] The basic components that form DNA strands are known as nucleotides or bases. There are four types of nucleotides - adenine ($A$), cytosine ($C$), guanine ($G$) and thymine ($T$). The arrangement of DNA in a cell nuclei maintains a form known as the DNA double helix. The double helix is formed by following the properties where $A$ bonds with $T$ and $C$ bonds with $G$ [51]. These pairs of bonds are known as complementary bases.[2]

Another type of molecule that can be observed in human cell nuclei is RNA. RNA is a nucleic acid that can be thought of as a string consisting of nucleotides: $A$, $C$, $G$ and the nucleotide Uracil $(U)$. In typical nuclei processes such as transcription and translation (see below) RNA is a single stranded molecule.[3]

[1] DNA also can be found in other cell organelles such as mitochondria and chloroplasts [56].

[2] A base-pair is denoted as **bp**, $10^3$ **bp** is known as 1 Kilobase (**kb**) and $10^6$ **bp** is known as 1 Megabase (**mb**)

[3] RNA can also exist in double stranded or folded forms [6].

#### 2.1.2.3 Molecular Implementations of Classical Genetic Concepts

Each chromosome in a cell consists of a DNA double helix. The entire DNA strand from the 23 pairs of chromosomes is known as the human genome [51, 56]. The region or locus of a single stranded DNA in a chromosome that encodes a protein is known as a gene. An allele corresponds to a particular DNA sequence for that gene. The number of genes in the human genome is yet to be confirmed but an approximate estimate tells us that the number of genes is between 80,000 and 100,000 [65].

Typically, in the double helix form of DNA in chromosomes, one strand is considered as the coding strand, where genes are expressed as proteins. Only a small fraction of DNA in complex organisms is expressed to form a protein [56]. The DNA regions that contain genes are implemented in protein in two steps - transcription and translation (see Figure 2.1($B$)). In the first step (transcription), the DNA coding strand in the gene region is used to produce a complementary RNA strand. In the second step (translation), this RNA strand is processed to form a protein.

### 2.1.3 Single Nucleotide Polymorphism (SNP)

Current genotyping technology gives a single genotype sequence that corresponds to the haplotypes inherited from the parents. The parents provide two alleles for each genotype position; if the parents provide the same allele, the position is in the homozygous state, and if these alleles are different it is in the heterozygous state. The alignment of the genotypes of any two persons will show nucleotide variation in some positions. These variations are known as *single nucleotide polymorphisms* (SNPs) (see Figure 2.2). SNPs are typically physically distant from each other by approximately

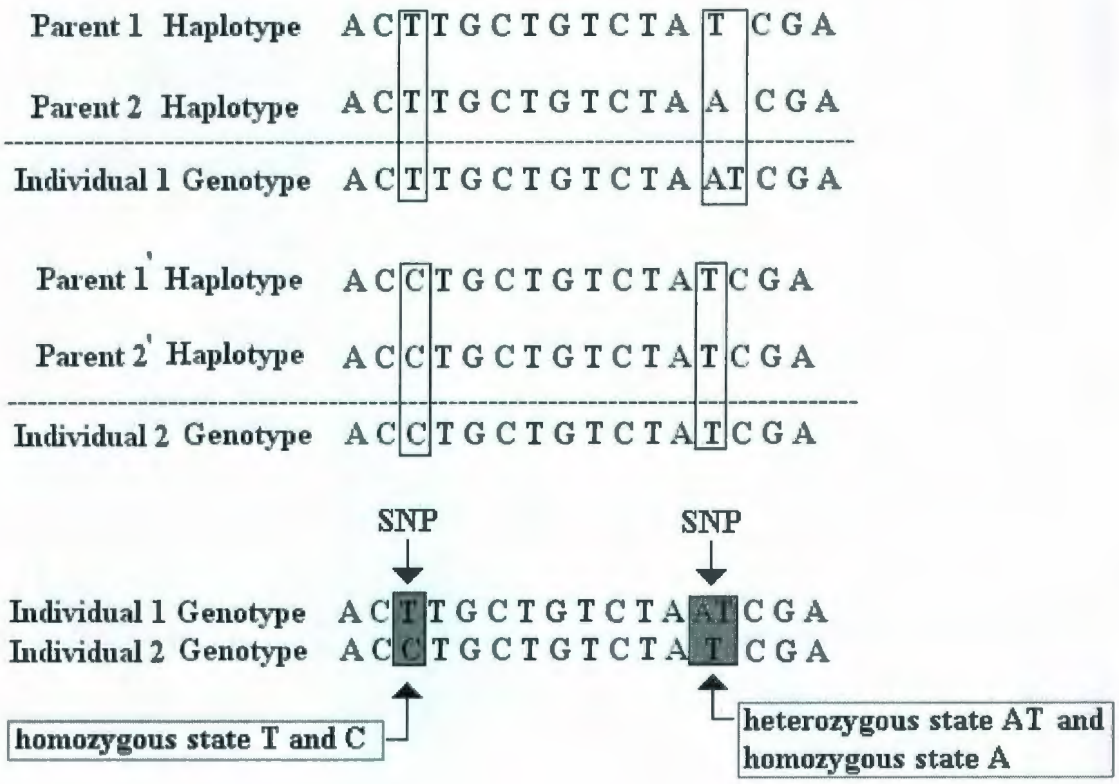


Fig. 2.2: SNP Variation in Two Individuals.

1000−1300 bases on DNA strands. Millions of SNPs have been successfully sequenced during the human genome project and the subsequent HapMap project for various analytical purposes. It has been observed that less than 1% of SNPs result in variation in protein. As of 2007, the International Hapmap Consortium has identified and mapped 3.1 million SNPs, and determining which of these SNPs have functional activities is currently a major area of research [58].

In this section, we will examine various aspects of SNPs that will be useful in this thesis, including general properties of SNP data (Section 2.1.3.1), linkage between SNPs (Section 2.1.3.2), haplotype pattern structure (Section 2.1.3.3) and problems with SNP data (Section 2.1.3.4).

#### 2.1.3.1 General Properties of SNP Data

Current genotyping methods produce SNPs in digital form and the properties of these digitized SNPs are important for any genetic analysis that includes SNPs. There are some basic established properties of SNP data that need to be considered before doing any types of analysis:

1. In any individual haplotype, each SNP position contains any two of the four nucleotides A,C,G, and T; these nucleotides are known as alleles.

2. In each SNP, the two alleles (see Section 2.1.3) are distinguished by their frequencies in a population. The major allele is the allele that is most frequent in that SNP position and the minor allele is the less frequent one.

3. The sequenced SNPs are linearly ordered according to their chromosomal position.

In Section 2.1.1, we described genotype and haplotype with regards to DNA. In regards to SNPs, the genotype and the haplotype need to be explained. For SNPs, a haplotype of an individual is a set of contiguous alleles that corresponds to the SNP positions [22]. Each individual has two haplotypes, inherited from that individual's parents. Haplotypes are also known as phase data. Popular sequencing technologies produce the two parental alleles for each of SNPs from the chromosomal region. This conflation of these two alleles for each SNP site is known as genotype or unphase data (see Figure 2.3). In complex disease association studies, haplotypes reveal more significant genetic variations than single SNP associations [10]. The conversion of a genotype into its associated pairs of haplotypes is a complex problem in computational

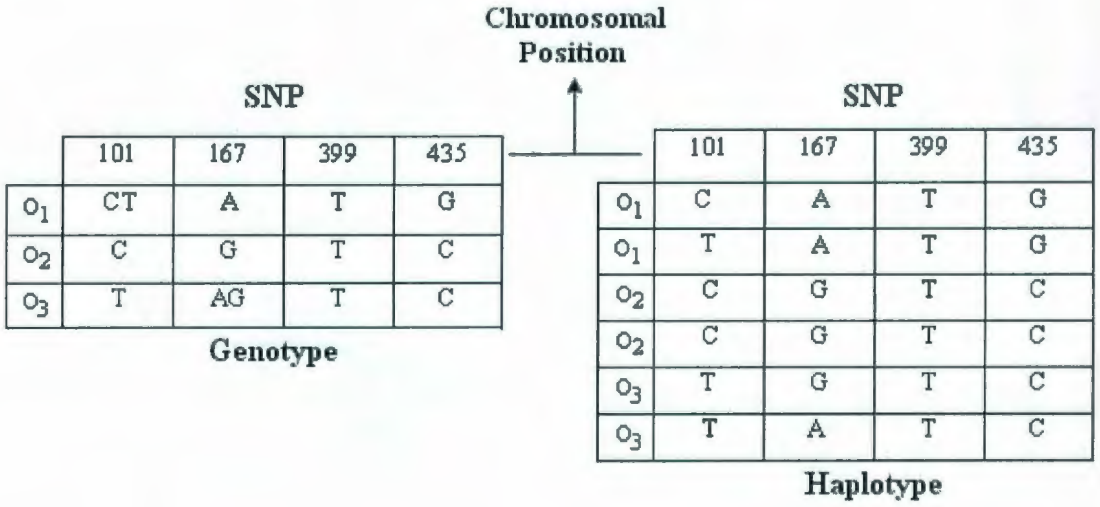


Fig. 2.3: Genotype and Haplotype Data.

biology and it will be discussed in Section 3.1 . In the remainder of this thesis, we will use the terms genotype and haplotype relative to SNPs.

#### 2.1.3.2 Assessing the Degree of SNP Linkage

Individual symbols in a mathematical sequence are typically assumed to be independent; however, SNPs in biological sequences are not independent. Groups of contiguous SNPs are often dependent and travel together as a block over generations.[4] This concept of SNP groups is known as linkage disequilibrium (LD). SNPs that are in high LD reside in close physical proximity on a chromosome, and such groups often contain a single gene or a set of closely related contiguous genes. The most commonly used measures of LD between a pair of SNPs are D′ and $r^2$(the computation detail of LD is given in Section 4.3.3). The respective range of D′ and $r^2$ is between 0 and

[4]The sizes of SNP groups that travel together in the human genomes is variable in length. The reasons for these variability are still unknown [3].

1. In terms of pairwise LD, $D' = 1$ (known as complete LD) means the LD between this pair of SNPs has not been disrupted by any biological facts (i.e. recombination) for generations [3, 43]. The $r^2$ measure defines the correlation of alleles in the SNP pair, such that where $r^2 = 1$ is known as complete LD.

#### 2.1.3.3 Haplotype Blocks and Haplotype Patterns

Haplotypes in the human genome have a block-like structure, so that a set of alleles from contiguous SNPs form a haplotype block if the SNPs are strongly linked. A set of SNPs that are strongly linked contains very few haplotype blocks. Recent studies have revealed this property of the haplotype blocks by examining different populations [14, 19]. Haplotype blocks can stretch as long as 100kb and this length differs in different populations.

In this thesis, we need to talk about a type of haplotype block consisting of alleles from non-contiguous SNPs. A haplotype pattern is a set of alleles that is obtained from a set of contiguous and non-contiguous SNPs that are linearly ordered in the genome. Complex disease analysis needs both haplotype blocks and haplotype patterns to locate underlying genes associated with SNPs [30].

#### 2.1.3.4 Problems with SNP Data

As noted earlier, we want to analyze haplotypes to investigate possible haplotype patterns related to a complex disease. However, due to technological limitations, the haplotype data may be problematic in two ways:

1. In the process of obtaining a genotype, there may be positions at which genotype data are missing.

2. Even given a complete genotype, known techniques for deriving associated haplotypes may leave certain positions unresolved.

Both missing and unresolved data are considered as missing data in the literature [24, 53]. The best method that constructs haplotypes from genotype data can have at most 20% of missing or unresolved data in the haplotypes [53]. The HapMap consortium implemented a quality control (QC) filter to map the human haplotypes. Their QC filter ignores genotype data if it contains $\geq$ 20% missing data [58]. This policy is simple to implement; however, ignoring missing data means a great loss of information for complex disease analysis. Hence, we have to handle missing data while investigating haplotype data. In Section 4.3.3, we propose an algorithm that can handle missing or unresolved data that arises in haplotypes.

## 2.2 Genetic Analysis of Human Disease

Though classical genetic analysis of human diseases concentrated on Mendelian disease, modern human genetics has shifted the focus towards the investigation of complex disease. With the completion of the human genome sequence in 2000, genome-wide scans for complex diseases became feasible. However, to analyze complex disease with the availability of large molecular genetic datasets, we have to adapt the classical genetic models of disease analysis.

In this section, we will discuss the various types of genetic data and the analytical models that use these data to isolate disease susceptible genes. In Section 2.2.1, a brief description of the different types of molecular data will be presented. In Section 2.2, we will discuss two basic analytical models that are used to investigate Mendelian

disease and complex disease.

### 2.2.1 Human Genetic Data

Genetic analysis for human disease uses a wide range of molecular data that includes DNA sequence, RNA sequence, protein sequence, gene microarray expression, micro satellite, copy number variation (CNV) and SNPs [56]. Each dataset has its own properties and the use of these data is dependent on the objective of the investigation. To determine the genetic basis of human disease, analysis mainly focuses on using DNA variation or SNPs for mapping human disease to specific genomic regions. The analytical models that use SNP data are discussed in the following sections.

### 2.2.2 Analytical Models

In genetic disease association studies, there are two basic types of analytical models that can be adapted: family-based linkage study and case-control association study. These are explained below.

A disease's history in a family can have a genetic basis. The genetic analysis of such diseases uses a family-based model. The family-based model dissects the genetics of complex disease at the individual level. Family members are genotyped according to their history of a disease, and these SNPs and haplotypes are analyzed to point out possible mutations in a family member that may be associated with that disease. This model of analysis is also known as linkage study. The data required for linkage analysis is hard to find; moreover, this investigation only focuses on finding the genetic structure of a complex disease relative to a particular family and may not

provide information for that disease relative to an entire population [10].

The case-control model dissects the genetics of complex disease at the population level. The set of all individuals examined as part of a case-control study is known as the cohort of that study. The case group in a cohort is the individuals that are diagnosed with a disease by a physician and the control group in that cohort is those individuals diagnosed with absence of the disease. The case-control model is useful in disease analysis because it points out the significant differences in occurrence of SNP alleles between the case and control groups. We have chosen to adapt the case-control model to investigate complex disease association with SNPs and their haplotypes because it has been proved that case-control association studies provide better and more consistent results than family-based studies. SNP cohorts used to analyze and detect haplotype patterns for a particular disease must satisfy the following properties.

- Each SNP in the control cohort must not deviate from Hardy-Weinberg Equilibrium (HWE[5].). Such HWE confirms that the genotype frequency distribution of an SNP in a cohort is stable or constant and that this distribution is not interrupted by any environmental factor [48]; hence, any deviation in the distribution of that SNP in the case cohort is probably associated with the disease being studied.

- The samples or individuals of the cohort must be taken from the same popula-

[5]HWE is a mathematically defined condition which states that the genotype frequencies in a population remain constant or are in equilibrium from generation to generation unless specific disturbing influences such as environmental factor or disease are introduced. A full mathematical description of HWE is given in [48].

tion (i.e. ethnically matched) because population stratification is a strong bias which can produce false positive results [11].

- The minor allele frequency of each SNP in a cohort must be $\geq 5\%$ because a SNP with low minor frequency ($< 5\%$) does not represent the two allelic frequency distributions in a population [56].

These assumptions must be maintained before applying any analytical algorithms,

### 2.2.3 Analysis of Mendelian Diseases

The analysis of genetic disease has traditionally focused on Mendelian disease. Mendelian diseases are diseases that are associated with a single gene [7]. Mutation in a Mendelian disease is usually a single nucleotide alteration in a gene which has an impact on the function of the associated protein. Over 1500 such genes are documented in the Online Mendelian Inheritance in Men (OMIM) database [7]. The primary investigation of Mendelian disease genes involves family-based analysis with SNP data.

### 2.2.4 Analysis of Complex Diseases

There is a wide variety of diseases that do not follow the Mendelian law of inheritance for disease because these diseases are regulated by a number of genes [10]. This type of disease is known as complex or multifactorial disease. The underlying genetic properties of complex disease have many open questions and to investigate these questions, the properties of Mendelian disease provide the basic building blocks for

solutions. The genes that cause a complex disease might have multiple mutations, where each mutation has an impact on protein function.

The common method of analyzing SNPs for complex disease in a case-control study is the single/multiple SNP window analysis [37, 36]. This is a two step process. In the first step, individual SNPs are analyzed by statistical tests for significant susceptibility to a disease. In the second step of the process, groups of larger and larger contiguous SNPs or multiple SNP windows are analyzed. In each window, the underlying haplotype blocks are tested to determine any significant association with respect to a disease. The same statistical tests may apply in both steps but they can also differ. In Chapter 3, previous work which has used these techniques to investigate complex diseases will be discussed in detail.

# Chapter 3

# Problem Formulation and Related Work

In Chapter 2, we reviewed the analytical models used to investigate Mendelian and complex disease. In complex disease investigation, single and multiple window analysis is the most commonly used technique, but these analyses are computationally expensive. It is important to find a computationally feasible way of investigating complex disease.

In this chapter, we examine the different computational approaches applied to investigate Mendelian and complex disease. In Section 3.1, we review computational problems that arise in human genetics and we formulate the haplotype pattern detection problem (HPD) examined in this thesis. In Section 3.2, previous work related to HPD is discussed. This section finishes with a list of requirements for an ideal algorithm for the HPD problem.

## 3.1 Computational Problems in Human Genetics

The great achievement of revealing the structure of DNA in 1953 opened doors to many new computational problems. Computational problems in genetics traditionally found on DNA sequencing, sequence alignment, protein folding and structure prediction [51]. In the last two decades, the molecular mapping between a gene and a disease (i.e. genotype phenotype relationship) was conferred only for those diseases that fall into Mendelian law or in other words, diseases that occur by a single gene [7]. The mapping of these Mendelian diseases refers to the SNP location in a chromosome that shows susceptible occurrences in the disease carrier group. Complex disease on the other hand, originate by multiple genes and the mapping of these genes is much more complex than that of Mendelian disease gene.

In Section 3.1.1, we will discuss the detection of susceptible single gene mutation for Mendelian diseases. In Section 3.1.2, computational problems for SNP and haplotype data with respect to complex disease are examined. The formal problem examined in the thesis, namely, haplotype pattern detection (HPD), is outlined in Section 3.1.3.

### 3.1.1 Detecting Genomic Regions for Mendelian Diseases

The two analytical models that were described in Section 2.2 are mostly used for the identification of disease-susceptible SNPs for Mendelian diseases. The computational problem here is to detect a disease-causal SNP from a set of SNPs that is genotyped from a set of individuals. These individuals that are genotyped might be used for a case-control model or the individuals might be genotyped from a family for linkage

model analysis. Two important success story of Mendelian disease gene detection are Cystic Fibrosis and breast cancer. Cystic Fibrosis causes breathing problem, respiratory infections and problems with digestion. The discovered Cystic Fibrosis gene CFTR (Cystic Fibrosis Transmembrane Conductance Regulator) is located on chromosome 7 [7]. The research was conducted on a partial pedigree from the Canadian population to locate the mutation on the CFTR gene. The classical linkage analysis technique was used to locate the mutation in that chromosome region. The BRCA1 gene is responsible for a fraction of breast cancer. Evidence suggests that breast cancer patients with an early age, have a mutation in the BRCA1 gene which is located on chromosome 17 [7]. In this case, the statistical risk ratio was computed from a partial pedigree.

Classical linkage analysis is the most prominent and successful of all methods for detecting mutations associated with Mendelian disease genes. This success of identifying genes and their mutations for a disease is possible because such single gene diseases obey the Mendelian laws of inheritance. It becomes problematic when the diseases do not follow the rules of Mendelian inheritance, as is the case in complex diseases.

### 3.1.2 Detecting Genomic Regions for Complex Diseases

The genetic properties of complex disease are not completely known at this point in time. It is an ongoing research initiative to unravel the basic genetic properties of complex disease. This research has led to a series of computationally challenging problems [23]. In particular, to gain better knowledge about genetic properties of

complex diseases, it is vital to learn the underlying structure of SNPs in the human genome. In Section 3.1.2.1, we will discuss various computational problems that arise in the analysis of complex diseases relative to SNPs and haplotypes. In Section 3.1.2.2, the formal description of the problem that this thesis investigates, namely haplotype pattern detection (HPD), is outlined.

#### 3.1.2.1 Deriving SNP Data

A genome-wide association analysis is both economically and computationally expensive. Researchers typically use LD information to reduce this expense (see Section 2.1.3). An SNP can be a proxy for a group of SNPs if they are all in complete or perfect LD. This SNP is known as a tag SNP. Finding a minimum set of tag SNPs is NP-hard [4]. There exist various approximation algorithms for selecting tag SNPs. The block-based model of finding tag SNPs is the most commonly used method. In a haplotype-block based method, an SNP is considered to be a tag SNP if it is in strong LD with a group of other SNPs [5]. The two commonly used LD measures D′ and $r^2$ are used to define strong LD. In the case of D′, the value must be $\geq 0.98$ and for $r^2$, it must be $\geq 0.80$ [19, 37].

Research on disease-correlated SNPs using single window analysis tend to focus on one gene while haplotype block mapping with diseases provides more insights about the disease susceptible-alleles of multiple genes [10, 14]. As stated earlier, the haplotypes for a subject reveal much more information than the corresponding genotype data. It is also known that haplotype association is much more powerful than single SNP association because it reveals the susceptibility of multiple genes corresponding to a disease [10, 37]. Separating two parental haplotypes from an individual's

genotype data is known as the haplotype phasing problem, and is known to be NP-hard [22, 24]. Different statistical and combinatorial approximation algorithms have been proposed to infer haplotypes from genotype data. Each of these methods has their pros and cons relative to accuracy. The two leading phasing algorithms are the PHASE and the EM-algorithm. The PHASE algorithm was found to be the most robust compared to all the other methods [33, 61]. This algorithm is a Bayesian approach that applies coalescent-based models to improve phasing accuracy. Even though it performs best among all the existing phasing algorithms, there can still be 20% missing or unresolved data in the phased haplotypes [53].

In our analyses, we will use haplotypes instead of genotype data because haplotype data is an important factor in the advancement of identifying disease associated genetic regions. Recently, various studies have revealed a very basic property of haplotypes in the human genome - namely, haplotypes with large block size have limited diversity in the human genome. Haplotype blocks can be 100kb in lengths and can contain multiple tag SNPs [19, 43]. This information is crucial for mapping haplotypes in the human genome. The mapping of a haplotype with a disease needs further computation after the phasing is completed. Haplotype block frequency estimation is one of the important aspects to assess disease association significance. Computing haplotype block frequency from genotype data is NP−hard [24], these same authors also provided an approximation algorithm based on maximum likelihood estimation to compute haplotype frequency. There are several variants of EM-algorithms that compute haplotype block frequencies [15, 33]. The PHASE algorithm mentioned earlier also computes haplotype frequencies.

#### 3.1.2.2 Detecting Haplotype Patterns

In the last two decades, computational problems in complex diseases have concentrated on localizing disease-correlated haplotype blocks to pinpoint the disease-associated alleles in the human genome. Most previous studies have used short haplotype blocks to find their susceptibility to a complex disease of interest. There can be one or more genes associated with each haplotypes block. It is not possible to investigate the disease susceptibility of different permutations of genes using the haplotype block method because this method only allows investigation of haplotype blocks obtained from contiguous SNPs. Hence, using the haplotype block method to investigate disease susceptibility may not be the best strategy [30]. To overcome this deficiency, we will formulate our problem in terms of haplotype patterns. Recall from Section 2.1.3.3 that a haplotype pattern is a set of alleles containing a contiguous or non-contiguous alleles from $n$ linearly ordered SNPs. In our problem formulation, we will focus on detecting such patterns in a case-control cohort.

A typical case-control cohort consists of $m$ case and $m'$ control individuals for a panel of $n$ SNPs in a chromosomal region. The SNPs are digitized from each sample using currently available genotyping technology. In this thesis, we will assume the haplotypes are obtained using a phasing algorithm which may produce missing data. The input is two matrices $M$ and $M'$ of case and control haplotypes, respectively, and we are interested in patterns that are significantly different between the two matrices. The haplotype pattern detection problem is formalized as follows:

HAPLOTYPE PATTERN DETECTION (HPD):

**Input:** Two matrices $M$ and $M'$ of the haplotypes over $n$ SNPs for $m$ and $m'$ individuals, respectively; where $M$ represents the case matrix and $M'$ represents the control matrix.

**Output:** A set of SNP patterns $P$ such that the frequency of each $p \in P$ is significantly different in both $M$ and $M'$ matrices.

Pattern significance is computed using statistical tests and is typically computationally expensive (see Section 4.3.3.3). This task is made even more challenging by the fact that, courtesy of limitations of current SNP genotyping technologies and haplotype reconstruction algorithms, 20% of the haplotype allele values in the given case and control matrices may be missing (see Sections 2.1.3.4 and 4.3.3.2).

## 3.2 Previous Work

In this section I will briefly review published research on solving complex disease problems with regards to SNPs and haplotypes. Most of the proposed approaches performed analysis of haplotype blocks (see Section 2.1.3.3). These studies can be categorized into three approaches, statistical, combinatorial optimization, and genetic algorithm, which are described in Sections 3.2.1, 3.2.2 and 3.2.3, respectively. The advantages and disadvantages of each approach are given at the end of each subsection.

### 3.2.1 Statistical Approach

The most popular methods for detecting haplotype blocks in case-control data use statistical models and tests. Some successful demonstrations of these methods for haplotype block association with a disease include Crohn's disease, Inflammatory Bowel disease (IBD), and Ankylosing Spondylitis (AS). All of these successful demonstrations used fixed-length haplotype blocks. After giving an overview of these studies and their associated statistical methods, we will describe some recent methodologies that allow variable length haplotype blocks.

One of the most well-documented statistical investigations was on Crohn's disease [13]. The investigation was performed on 258 cases and an equal number of ethnically-matched control samples for a panel of 103 SNPs that spans a 500kb region on chromosome 5q31 [47]. All SNPs were tested and excluded if they showed any deviation from the Hardy-Weinberg Equilibrium (HWE) or if the minor allele frequency is $< 5\%$. The authors developed a hidden Markov model (HMM) based on LD measure D' to capture fixed length haplotype blocks with higher frequency. The genotypes were phased into haplotypes using the GENEHUNTER application which uses an EM-based algorithm [14]. This EM-based algorithm can handle missing data by computing a maximum likelihood estimation to compute the probability of that missing genotype. After phasing the genotype data into haplotypes, the haplotype frequencies were computed simply by counting.

Another successful investigation was conducted on inflammatory bowel disease (IBD), which is a chronic inflammatory disorder. In this research, the case-control cohort was genotyped from the German population for 33 SNPs on chromosome 10q23

[55]. This panel of SNPs contains a set of genes that spans a 5MB region. All SNPs were tested for HWE, and 28 SNPs were analyzed by calculating $\chi^2$ values and using Fisher's exact tests. The odd ratio was calculated using Fisher's contingency table. The permutation technique was also incorporated to see the $\chi^2$ effect in a more general population: 100,000 permutations were performed on the set of 28 SNPs, and single $\chi^2$ value greater than 9.91 was considered to define a significant $p$-value. GENEHUNTER application was used to accommodate these statistical tests into the investigation. The results showed that two haplotypes consisting of 18 markers in the DLG5 gene showed strong susceptibility to IBD.

Inflammatory Arthritis is one of the most common complex diseases in any population. Arthritis has different variants one of which is Ankylosing Spondylitis. Maksymowych *et al.* [37] conducted single window and 3 window haplotype block association tests on three Canadian populations (Alberta, Newfoundland and Toronto). The authors genotyped 38 SNPs on chromosome 2 for each of the three case-control cohorts. This chromosomal region spans 360kb and includes the IL1 gene cluster. Eight SNPs were removed because of $< 5\%$ minor allele frequency and 1 SNP showed deviation from HWE. Groups of SNPs with strong LD were reduced to single SNPs, resulting in the removal of 9 SNPs. A panel of 20 SNPs was analyzed for haplotype association. The 3 window haplotype block association tests were performed by using the application WHAP [46]. The authors found haplotype blocks with significant correlation with AS on 8 consequential windows. There are 9 haplotype blocks, each with three consecutive alleles found in these 8 windows, that are correlated with the AS disease and include IL1A, IL1B and IL1F7 gene. The haplotype phasing was performed by applying an EM-based algorithm. For each window, an omnibus statistical test

was performed by applying 10,000 permutations, and the global significance of each $p$-value was determined by permuting the data.

Recently, some research has attempted to break the haplotype block fixed-length barrier by allowing variable block length. Browning [9] proposed a statistical model which used a variable length Markov chain to detect variable length haplotype blocks. He relied on two different phasing algorithms to obtain haplotype data and assume that there is no missing data. In his approach, each chain represents a haplotype block with contiguous alleles. The Fisher exact test was used to obtain significant $p$-values for each haplotype. The algorithm was tested on two previously published case-control datasets for Cystic Fibrosis and Crohn's disease, and the previously published results for both datasets were in strong accordance with his findings.

Another study [30] proposed a regularized regression analytical model allowing variable length haplotype blocks. The authors assumed that haplotype data are given that contain no missing data. For each haplotype block, the significant $p$-value was obtained by using a Fisher exact test. The authors tested their proposed methods on multiple simulated datasets and one real datasets for Parkinson's disease. They have showed that their proposed method performs consistently when compared with the other proposed methods.

Though the statistical methods described above are preferred because they both incorporate extensive biological constraints and are based on proven older techniques, they are exceptionally computationally expensive, often taking on the order of months to run, and can only compute haplotype blocks consisting of a few adjacent SNPs. Even if the computational effort associated with the biological constraints can be tamed, efficient algorithms for finding optimally significant haplotype patterns com-

posed of non-adjacent SNPs probably do not exist. Moreover, regarding the missing data problem, both GENEHUNTER and WHAP use different maximum likelihood estimations to handle missing data in the genotype data, and do not incorporate any biologically meaningful approach or knowledge of the genetic properties of the dataset to handle missing data.

### 3.2.2 Combinatorial Optimization Approach

Little work has been done on combinatorial optimization methods for detecting haplotype blocks or patterns for complex diseases.[1] Most of this work has focused on developing algorithms for haplotype block frequency computation. Halperin and Hazan [24] showed that computing haplotype block frequency from genotype data is NP-hard and proposed an approximation algorithm. They have also shown that haplotype block frequency computation from haplotype data takes polynomial time. In their approach, they include a probabilistic technique to compute haplotype block frequency from haplotype data with missing values.

On examining the literature, there appears to be only one combinatorial optimization paper that addresses a problem remotely like the haplotype pattern detection (HPD) problem. Yosef *et al.* [69] investigated genotype patterns that distinguish case individuals. In their problem formulation, the genotype data was not converted into haplotype data and the case-control model was extended to accommodate multiple phenotypic individuals instead of controls. The authors formulated the problem as below:

[1]Little work has also been done for to detect genotype blocks for complex diseases. See [60] for an overview of this work.

DISCRIMINATING PATTERN PROBLEM (DPP):

**Input:** Given a bipartite graph $G = (P, F, E, w)$ with w: P$\rightarrow\{-1, 1\}$.

**Output:** A feature subset $F' \subseteq F$ such that the biclique defined by $F'$ has maximum summed vertex weight.

Here, $P$ denotes the population under study, $F$ is the set of all feature states (or SNPs) and $E$ is the edge set that connects each individual to the feature states it possesses. The authors construct a graph and assigned weights ($+1$ for case,$-1$ for other phenotypes) to the vertex set. The authors proved that DPP is NP-hard. The authors also implemented a heuristic algorithm and the performance of the algorithm was verified using both simulated and real data. Unfortunately, this heuristic was not verified against other established methods (i.e. statistical methods), and missing genotype data was ignored during the construction of the graph.

The combinatorial optimization approach is appealing because there is a very large literature on combinatorial optimization which has potential application to problems like HPD. However, it is crucial for any combinatorial optimization technique to incorporate biological constraints and handle missing data. Hence, the combinatorial approach in the detection of haplotype patterns in a case-control cohort requires more attention and effort.

### 3.2.3 Genetic Algorithm Approach

Some work has been done on detecting variable-length haplotype blocks using a heuristic technique for solving combinatorial optimization problems called genetic algorithms (GA). Nakamichi *et al.* [39] used a standard genetic algorithm to detect a set of SNPs and their correlation with environmental factors (i.e. age). Their technique focused on capturing individual significant SNPs rather than haplotype blocks. The genetic algorithm genotype representation was a variable length vector consisting of SNP alleles. The fitness function was designed based on logistic regression and the Akaike Information Criterion (AIC). The AIC measure provides analytical power to detect a set of SNPs that are most correlated with a disease of interest. The algorithm was executed on a real dataset (96 cases with diabetes and an equal number of healthy controls). These individuals were genotyped for 720 SNPs with age as the environmental factor. The GA results were not compared against any other computational model. The authors found 7 SNPs that showed significant association with the environmental factor in the disease group. Their proposed algorithm does not take missing data into account while computing SNP significance.

Clark *et al.* [12] designed a standard genetic algorithm to detect haplotype patterns that are in strong LD in the case group. The chromosomal region examined is assumed to be susceptible to a disease of interest; hence, only case groups are investigated to locate the haplotype patterns that are in high LD. In this particular scheme, a genetic algorithm logic tree (using OR and AND operators) was used as the representation and each tree was constructed in such a way that a set of patterns can be derived from (and hence are associated with) that tree. The mutation and crossover

operators of the genetic algorithm are performed based on LD value between pair of SNPs. The LD values between multiple SNPs in a logic tree was calculated using the D′ measure. They tested their algorithm on a Nigerian population containing 738 case samples with hypertension. A set of 13 SNPs were genotyped that span a 26kb region on chromosome 17. The region showed 6 SNPs that are in strong LD in the case groups. The authors assumed the haplotype data was complete and did not have any missing data.

Genetic algorithms have the potential, by manipulating the fitness function and representation, to integrate biological constraints and handle missing data. However, the two genetic algorithms discussed above are not designed for detecting susceptible haplotype patterns from a case-control cohort and neither algorithm handles missing data.

The review of advantages and disadvantages of the three approaches discussed above gives us two requirements for an ideal computational method for detecting haplotype patterns in case-control data. The first requirement is to detect haplotype patterns instead of haplotype blocks from a case-control cohort. The second requirement arises while computing haplotype pattern frequency if data is missing - namely, it is important to include genetic properties of SNP data (especially knowledge of LD) when handling missing data. Recent studies have revealed the block-like structure of haplotypes using LD information of SNPs and it is crucial to handle missing data to find this block structure. Similarly for complex disease association analysis, handling missing data in haplotypes is crucial because datasets are expensive to obtain, and large datasets are needed to obtain power in statistical tests, ignoring missing data is not helpful [50].

Detecting haplotype patterns considering contiguous and noncontiguous alleles makes the search space enormous. Exhaustive search is not practical for a moderate size of data. Hence, it is important to adapt a fast search technique. Genetic algorithms offer such a fast search technique. In Chapter 4, we will discuss the basics of genetic and cooperative coevolutionary genetic algorithms (CCGA) and will propose a CCGA for haplotype pattern detection in case-control haplotype data.

# Chapter 4

# Algorithm Design

In the previous chapter, we have discussed the advantages and disadvantages of different approaches that have been proposed to detect SNPs and their underlying haplotypes susceptible to a disease. In this chapter we will present an algorithm that encompasses the advantages and alleviates the disadvantages.

In the last few decades, genetic algorithms have been used to solve various complex problems with promising results. In Section 4.1, we will discuss the basic components of a standard genetic algorithm and the mechanisms executing the genetic algorithm. In Section 4.2, we will outline a variant of genetic algorithms called cooperative coevolutionary genetic algorithms (CCGA). The proposed CCGA to solve the HPD problem will be presented in Section 4.3.

The standard terminology for genetic algorithms reuses many of the terms from classical genetics. There is a potential for confusion. When it is obvious in context whether we are referring to biological entities or genetic algorithm entities, we will just say the term, eg. gene, chromosome. However, if it is not clear from the context,

Standard Genetic Algorithm

1. $gen = 0$

2. randomly generate initial population $P(gen)$

3. while $(gen \leq max\_gen)$

4. select parent chromosome from $P(gen)$ and apply genetic operators

5. evaluate fitness of each chromosome in $P(gen)$

6. select chromosomes from $P(gen)$ for next generation

7. $gen = gen + 1$

end while

Figure 4.1: Pseudocode of a Standard Genetic Algorithm

we will put that context in front of the term, eg. genetic algorithm chromosome, biological chromosome.

## 4.1 Standard Genetic Algorithm

Genetic algorithms is a computational model that was inspired from the theory of evolution in biology. Holland in 1975 proposed the theocratical adaptation of the evolutionary theory and showed how it could be applied to solve computational problems [25]. In the last few decades, extensive work has been done on the theory and application of genetic algorithms. There are complex problems where genetic algorithms are shown to outperform various types of proposed deterministic heuristics [38, 67].

Pseudocode for a standard genetic algorithm is given in Figure 4.1. This algorithm has four basic components to facilitate its evolutionary process. These four components are integrated to search for problem solutions. These components are

*population, genetic operator, fitness evaluation,* and *selection* [25]. A standard genetic algorithm contains a population of individual chromosomes. Two basic types of genetic operator, *crossover* and *mutation*, are used to modify individual chromosomes to produce offspring. The *fitness* of the modified chromosomes are then evaluated. The *fitness* evaluation is based on a fitness function that is designed to solve a given problem. The next component, *selection*, chooses those chromosomes from the current population that are highly fit to produce offsprings for the next generation. This process of modification-evaluation-selection executes for a certain number of generations until the termination criterion is met.

### 4.1.1 Population

The basic element in a population is a chromosome. In a standard genetic algorithm, there is usually one population containing a certain number of chromosomes. Let $p$ denote a chromosome and $i$ denote the index of the chromosomes in a population. Each chromosome $p_i$ can be represented as a vector of binary bits (or any other data type) with a length $l$ (see Figure 4.2) and the length of each chromosome is problem dependent. These chromosomes in a population evolve for a number of generations to produce solution for a target problem. The knowledge of the problem is the key to determine the length of a chromosome and the size of a population. Most genetic


Fig. 4.2: A Binary Vector Representation of a Genetic Algorithm Chromosome.

algorithms design performs sensitivity analysis to decide the population size for a problem. There is strong evidence suggesting that population size is one of the most important parameters in the process of evolution [70, 29].

There exists a popular genetic algorithm variant called steady state genetic algorithms where parent and offspring compete with each other to win their position for the next generation [52]. In contrast to generation-based genetic algorithms, steady state genetic algorithms maintain a replacement strategy that defines which members of the population will be replaced by the new offspring. Hence, in a steady state genetic algorithm, the chromosomes are selected from the parents and the offspring for the next generation.

### 4.1.2 Fitness Function

The concept of a fit individual is complex in nature, and there is no straightforward way to quantify that an individual is more fit than others. However, in computational models, we can certainly quantify the fitness of individuals in the population according to the target solution criteria. A fitness function $f$ is an objective function that quantifies the optimality of a chromosome in solving a problem [38]. Fitness is the driving force of the evolutionary search process, in that the fitness value of a chromosome determines whether its genetic materials will be carried over to the following generation.

The design of the fitness function affects the overall performance of a genetic algorithm. While designing a fitness function, one should be cautious about the fitness landscape which is derived from the representation, as this landscape may

cause the fragmentation of the search space such that there are many local optima. In an ideal scenario, the fitness landscape should be smooth for genetic operators to climb to the optimum solution.

### 4.1.3 Genetic Operators

Genetic operators are basic mechanisms to explore the search space and to maintain diversity in a population. There are two types of operators that are frequently used to produce offspring for new generations - crossover and mutation.

A crossover operator exchanges chromosome segments between two or more individuals to produce an offspring for the next generation. Crossover does not introduce new information into the offspring chromosome but rather exploits the search space using information from fit individuals [68]. Different variants of crossover exist but the most commonly used ones are *one-point crossover*, *two-point crossover*, and *multi-point crossover*. More variants of crossover can be customized for any problem.

One-point crossover operates on a pair of parental chromosomes where a random point is selected and the segments of the two parental chromosomes are swapped to produce one or two offspring (see Figure 4.3). Crossover rate is a parameter that determines the probability to perform crossover on parents. The crossover rate usually set by the designer of the GA and the typical rate is around 80 to 100 percent. Crossover rate is an adjustable parameter and the adjustment depends on the overall design of the GA.

Mutation introduces new elements to a chromosome and is able to shift the population to search the space in a different locality. Mutation is a genetic operator that

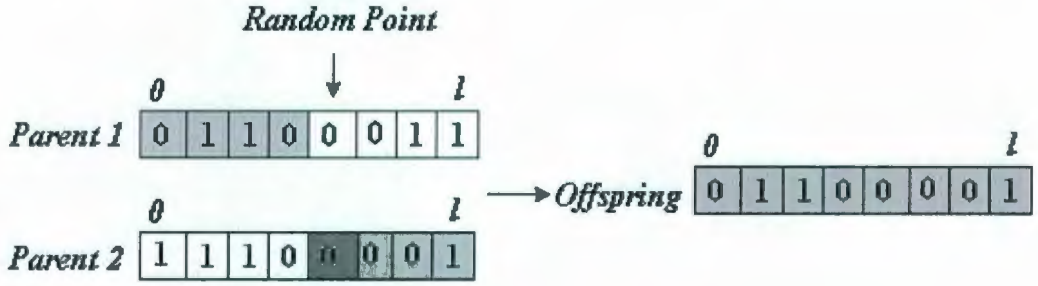


Fig. 4.3: A One-Point Crossover Operation on Two Parents.

alters one or more chromosomal values in a chromosome from its initial state. With these modified chromosomal values added to the population, the genetic algorithm may be able to reach a better solution which was previously not possible. This is the case, in problems which have local optima in the search space, where populations may prematurely converge to sub-optimal solutions. Mutation is an effective operator that helps populations to escape from local optima.

There are multiple variants of mutation operators, and among them *one-point*, *two-point*, and *multi-point mutation* are the most commonly used [68]. The one-point mutation operator randomly selects a position in the parent chromosome and alters the chromosomal value at that position. The application of a mutation operator is determined by the mutation rate parameter. Unlike crossover rate, mutation rate is usually low and it varies mostly between 0.001 and 5 percent. One reason to keep the mutation rate low is because a high mutation rate might disrupt good building blocks and interfere with the evolutionary process of the algorithm. Figure 4.4 gives an example of the one-point mutation operation.

The search capability of a genetic algorithm depends on the design of its genetic operators. Crossover uses the inheritance mechanism to exploit fit chromosomal seg-

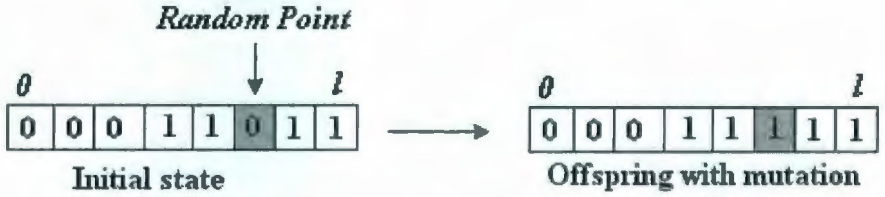


Fig. 4.4: A One-Point Mutation Operation.

ments. These inherited chromosomal segments are the building blocks for a possible more fit solution. Mutation helps to prevent the population from stagnating at any local optima. In the situation where a population has converged into a local optima and crossover cannot produce a solution by exploiting its parental chromosomes, mutation adds new information to help the population escape the local optima.

### 4.1.4 Selection

The concept of selection was adapted from Darwin's natural selection [25]. The selection mechanism operates on a population of chromosomes and it can be applied at two different stages of a genetic algorithm: *parent selection* and *survivor selection.* Parent selection is used to decide the individuals on which genetic operators such as crossover and mutation will be operated. Survivor selection is used to decide which chromosomes will be carried over to form a population for the next generation.

There are different types of selection mechanisms, of which *tournament*, *rank*, and *roulette selection* are the most commonly used [21]. To have a better understanding of how a selection process works, we will discuss the tournament selection technique. A tournament selection with a tournament size of 2 is a technique where a pair of chromosomes is selected randomly and then they compete with each other to win the

tournament. Selection pressure can be adjusted by altering the size of the tournament. The larger the tournament size, the stronger the selection pressure.

## 4.2 Cooperative Coevolutionary Genetic Algorithm (CCGA)

The concept of CCGA was first introduced by Potter *et al.* [45]. The authors proposed this algorithm by undertaking substantial modification of the standard genetic algorithm. The main distinction between a standard genetic algorithm and a CCGA is that the latter simultaneously evolves multiple populations where each population evolves a sub-solution for a target problem. Comparison of CCGA with other genetic algorithms has shown that CCGA gives better performance for various complex problems [31]. A CCGA has four different components – *species*, *genetic operators*, *collaborations and fitness evaluation* and *selection*. In the following subsections, each of these components are discussed.

### 4.2.1 Species

Unlike the standard genetic algorithm which maintains a population with multiple chromosomes, a CCGA maintains multiple populations, each of which is called a *species*. The chromosome in each species is known as a member. The species are separated based on the decomposition of the problem and the species should not overlap with each other in their search space. The idea of cooperation is implemented in CCGA by combining members of different species into one chromosome [45]. This

```
Cooperative Coevolutioanry Genetic Algorithm
1. gen = 0
2. for each species S
3.     randomly generate population P_S(gen)
4. while (gen ≤ max_gen)
5.         for each species S
6.             select parent chromosome from P(gen) and apply genetic operators
7.             evaluate fitness of each chromosome in P_S(gen)
8.             select P_S(gen) for next generation
9.         gen = gen + 1
   end while
```

Figure 4.5: Pseudocode of a Cooperative Coevolutionary Genetic Algorithm

one chromosome is the solution to the target problem.

Each species in a CCGA evolves in its own search space. In this way, the search process gets an edge to exploit each partition of the search space simultaneously instead of tackling the entire search space like the standard genetic algorithm. Since each sub-search space is smaller than the entire search space, CCGA may find better solutions faster than standard genetic algorithms [28].

### 4.2.2 Collaboration and Fitness Function

Collaboration of members from different species is one key difference between CCGA and the standard genetic algorithms. Each member in a species is a possible subcomponent of a solution. The fitness of each member is evaluated based on how well it

collaborates with members in other species to solve the entire problem. Prior to the fitness evaluation, a member in a species needs to combine with members in other species to form a solution to the given problem.

Collaboration can be implemented in different ways. De Jong *et al.* proposed two basic types of collaboration, namely, *random member collaboration* and *best member collaboration* [28]. For both collaboration methods, in each generation each species provides a member which is called a *representative* of that species. In random member collaboration, the representative is chosen from a species randomly. In best member collaboration, the fittest member of each species is chosen as the representative. Given that, each member of a species is combined with the provided representatives of other species to form a solution. The fitness of the solution strictly becomes the fitness of the member and is not shared with representatives that participated in the collaboration. The authors of [44] constructed a CCGA incorporating the two collaboration methods to solve the same problem, and reported that best member collaboration outperforms the random member collaboration technique for certain problems.

### 4.2.3 Genetic Operators

CCGA apply the two operators, crossover and mutation, like standard genetic algorithms. The main difference is that the operators only apply to members in the same species and inter-species genetic operation is not allowed. Hence, a crossover and mutation operation must always pick members from the same species. Since the crossover and mutation operators work within the same species, the exploitation and exploration of the fitness landscape are carried out locally [28]. This process of

applying genetic operators in each species helps to build better sub-solutions which also improves the quality of the combined solution.

### 4.2.4 Selection

As with genetic operators, selection in a CCGA also takes place inside each species and selection is independent for each species. Hence, the selection pressure of one species does not affect the evolutionary process of another species because the selection pressure applies to local members of a species to direct the evolution of that species [45].

## 4.3 CCGA for Haplotype Pattern Detection

In the previous two sections, the basic layout of a standard genetic algorithm and a CCGA have been given. It has been shown that CCGA are much more efficient than standard genetic algorithms for complex optimization problems [44]. This motivates our investigation of the applicability of a CCGA to the HPD problem. The HPD problem that was described in section 3.1.2 is an optimization problem with a large search space, due to the large dimensionality of the SNP dataset. There is no genetic algorithm or CCGA, to our knowledge, that solves the general version of the HPD problem.

The schematic diagram of our proposed CCGA scheme is given in Figure 4.6. Each of the components of the flowchart is described in the next sections. In Section 4.3.1, the general algorithm for the HPD problem is discussed. The species structure for the proposed CCGA is described in Section 4.3.2. The collaboration mechanism

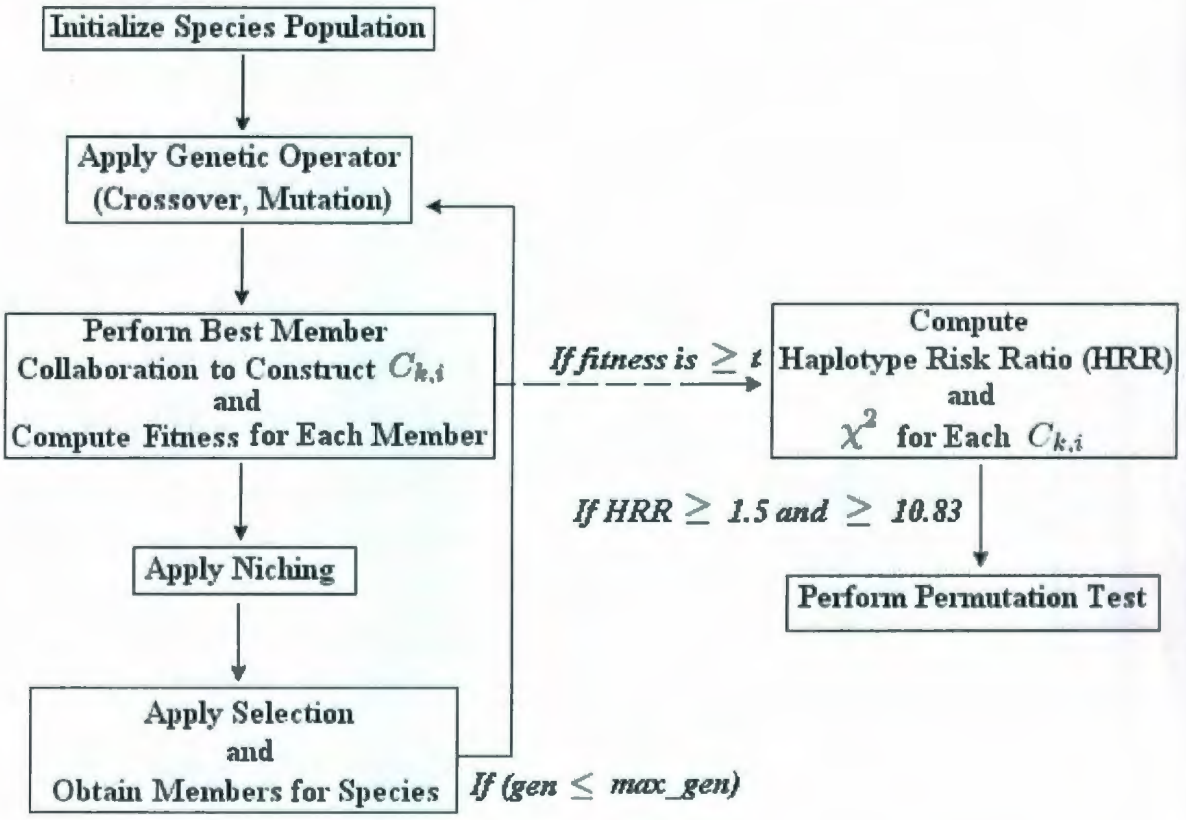


Fig. 4.6: Schematic Diagram of Haplotype Pattern Detection (HPD) CCGA.

and the fitness evaluation are discussed in Section 4.3.3. Finally, the genetic operators and the selection mechanism for the proposed CCGA scheme are given in Sections 4.3.4 and 4.3.5, respectively.

### 4.3.1 General Algorithm

Recall from Section 3.1.2 that the HPD problem input consists of a case matrix with $m$ individuals and a control matrix with $m'$ individuals for a set of $n$ SNPs. The columns of the matrices $M$ and $M'$ represent SNPs. Meanwhile, each column position contains major and minor alleles for a SNP and each pair of rows represents the

haplotypes of an individual with $n$ SNPs.

To understand the size of the problem search space, it is important to specify the key parameters of the search space. The search space depends on three key variables, $m$, $m'$, and $n$. Although there are $m$ and $m'$ individuals in matrix $M$ and $M'$, respectively, the total number of rows of the matrices are $2m$ and $2m'$, because each individual is represented by 2 rows of alleles, one from each parent. Each SNP has two alleles, major and minor; hence, each column of the given case-control matrices can be represented with 1 denoting the major allele and 0 denoting the minor allele. A column position where the allele is missing is represented with "$-$". Let $U$ denote the total search space which includes the search space of both matrices, hence $U = M \cup M'$. The size of the set of all haplotype patterns with contiguous and non-contiguous SNPs in $U$ is at most

$$\leq (2m \times 2m') \sum_{i=1}^{n} (3^i) - 1 \leq O(3^{n+1} max(m, m')) \tag{4.1}$$

The problem of searching for haplotype patterns with maximum frequency differences from the defined set in Equation 4.1 is a multimodal problem because there can exist more than one significant peak in the fitness landscape. In other words, there can exist a set of haplotype patterns instead of one with maximum frequency difference between case and control matrices.

The pseudocode for our CCGA is given in Figure 4.7. Detailed descriptions of the key steps, namely collaboration and fitness computation (step 12), HRR computation and permutation test (step 15), niching (step 16) and selection (step 17) are given in the following subsections.

A CCGA for Haplotype Pattern Detection:

1. $gen = 0$
2. for ($a = 1$ to $S$)
3. for ($b = 1$ to $k$)
4. initialize $P[a][b]$;
5. while ($gen \leq max\_gen$)
6. for ($a = 1$ to $S$)
7. for ($b = 1$ to $k$)
8. perform a **one-point crossover** to obtain a Offspring $O[a][b]$
9. if *(mutation == true)*
10. perform a **one-point mutation** on $O[a][b]$;
11. for each species pick best the members from $P$ and $O$
12. perform **best member collaboration** to form a complete solution $C_{k,i}$ and **compute fitness** for each $C_{k,i}$
13. for (each such $C_{k,i}$)
14. if ($f(C_{k,i}) >= t$)
15. compute **haplotype relative risk (HRR)** and perform **Permutation Test**
16. apply **niching**
17. perform **selection**
18. $gen = gen + 1$;

end

Figure 4.7: Algorithm Pseudocode for HPD CCGA.

### 4.3.2 Species

Knowledge of the HPD problem helps to design species that evolve sub-solutions to promote the discovery of the entire solution. In particular, the search space of the problem is heavily dependent on the number of SNPs. As shown in Equation 4.1, the size of the search space rises exponentially when the number of SNPs increases. Hence, in this CCGA scheme, the number of species $k$ is determined by considering the number of $n$ SNPs. Let $S$ denote the set of species. Equation 4.2 ensures that each species $S_x$, where $x \leq k$, will have 10 SNPs except the last species $S_k$, which will add the remaining SNPs if the remaining SNP is less than 3. Otherwise, an extra species will be added to contain the remaining SNPs. To maintain the linear order of the SNPs, the first species will contain the first 10 SNPs, the second species will contain the next 10 SNPs and so on (see Figure 4.8).

$$k = \begin{cases} n/10 & \text{if } (n\%10) < 3 \\ n/10 + 1 & \text{if } (n\%10) \geq 3 \end{cases} \tag{4.2}$$

After decomposing the SNPs into different species, the member initialization takes place within the vicinity of the allocated SNPs for a species. Let $l$ denote the number of SNPs on which each species $S_x$ is based. Each member $p_{x,i}$ in species $S_x$ is a vector with length $l$ consisting of a major or a minor allele in each position. This major and minor allele can be represented by 1 and 0, respectively. To detect a haplotype pattern in a given biological chromosomal region, a mechanism is required that will allow the investigation of haplotype patterns derived from contiguous and non-contiguous SNPs. In our haplotype pattern representation, a don't care bit "$*$" is considered along with the major and minor allele to satisfy this requirement. Hence,

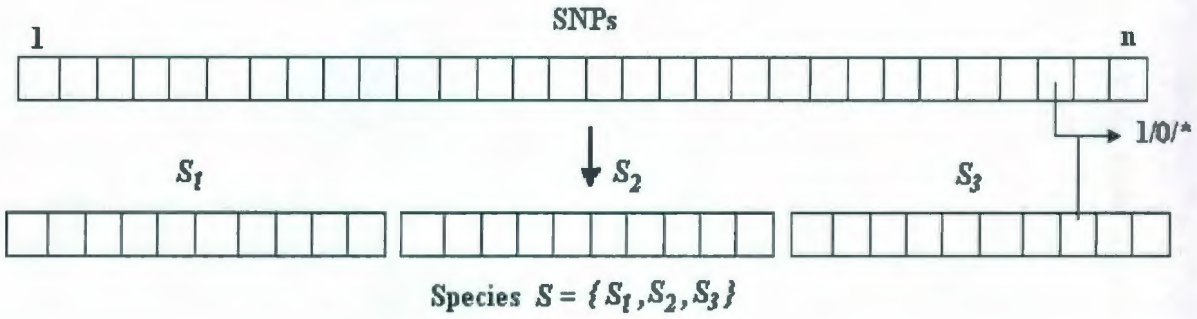


Fig. 4.8: Decomposition of Length-$n$ SNP vector into $k = 3$ Species.

each member $p_{x,i}$ of a species in our CCGA is a vector where each position contains a 1 or a 0 or a $*$ (see Figure 4.8). Each member $p_{x,i}$ represents a haplotype pattern over $l$ SNPs and the collaborated solution is a haplotype pattern over $n$ SNPs. The collaboration mechanism and the fitness evaluation are explained in the next section.

### 4.3.3 Collaboration and Fitness Function

The decomposition into species and the interaction between these species are important to the performance of a CCGA scheme. In the Section 4.2.2, two types of collaboration methods have been described, random member collaboration and best member collaboration. In this thesis, I have not compared these two collaboration methods; rather, I have used knowledge of molecular genetics to decompose the population into multiple species and decided to use the *best member collaboration* method in our HPD CCGA.

Collaboration takes place before the fitness evaluation. During the first generation of each species when all individuals were randomly generated, it is not possible to apply best member collaboration because no member has an assigned fitness. Hence,

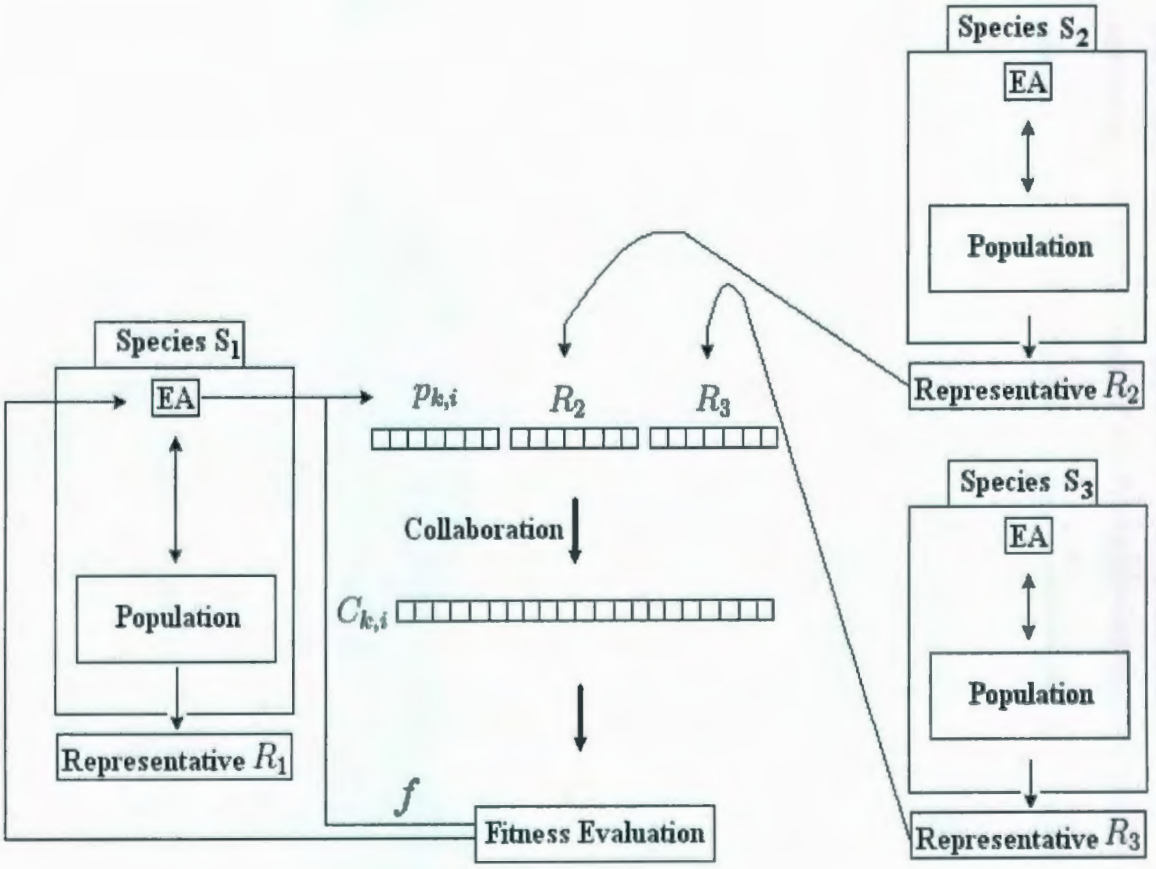


Fig. 4.9: Collaboration in CCGA Model with $k = 3$ Species.

random member collaboration was performed in the first generation. Consequently, the best member collaboration technique is adopted where at each generation, one representative with the best fitness is selected from each of the $k$ species. These representatives are combined with members in other species to form a possible solution for fitness evaluation. Let $R = R_1, R_2, ...., R_k$ be the set of representatives from each species. Each member $p_{x,i}$ of a species is combined with the representatives of other species to construct a solution $C_{x,i}$ for fitness evaluation. The evaluation result becomes the fitness of $p_{x,i}$.

As an example, for a set of $k = 3$ and species $S = \{S_1, S_2, S_3\}$, there can be a set of best fitness representatives $R = \{R_1, R_2, R_3\}$. A member $p_{1,1}$ from species $S_1$ will collaborate with representative $R_2$ and $R_3$ to construct $C_{1,1}$ for fitness evaluation with function $f$. A graphical illustration of this example is given in Figure 4.9.

As stated previously, the total SNP set was decomposed by maintaining its linear SNP order and it is important to maintain this linearity when collaboration takes place. Hence, a single solution $C_{x,i}$ with length $n$ is formed by aligning the vectors of each member $p_{x,i}$ sequentially according to their species number. We can formalize the collaboration as follows:

1. In the first species $S_1$, the collaboration combines the member $p_{1,i}$ with representatives from other species sequentially $\{p_{1,i}, R_2, ...., R_k\}$to form $C_{1,i}$ .

2. After each member of $S_1$ participates in the collaboration, the collaboration of the second species starts by combining the member $p_{2,i}$ with other species representatives sequentially $\{R_1, p_{2,i}, R_2, ...., R_k\}$to form $C_{2,i}$ .

3. This process applies to all other species.

The fitness of each $p_{x,i}$ is calculated by applying the collaborated solution $C_{x,i}$ to the following fitness function,

$$f(C_{x,i}) = |fr(M, C_{x,i}) - fr(M', C_{x,i})| \tag{4.3}$$

Equation 4.3 is a function which takes $C_{x,i}$ as an input and produces the frequency difference of $C_{x,i}$ in matrices $M$ and $M'$. The function $f(C_{x,i})$ returns a value between 0 and 1 which is the absolute difference of the two frequencies and this value is the assigned fitness for the participating member $p_{x,i}$.

In the following sub-sections, we will discuss how the frequencies are calculated from the two matrices for each collaborated single solution $C_{x,i}$. The frequency computation of a haplotype pattern is given in Section 4.3.3.1. In Section 4.3.3.2, the algorithm for handling missing data when computing haplotype pattern frequency is given. The description of haplotype relative risk (HRR) and its statistical significance for each haplotype is discussed in Section 4.3.3.3. Detail of the Permutation test for each haplotype that passes the HRR significance test are also described in this section. Finally, the niching criteria are given in Section 4.3.3.4.

#### 4.3.3.1 Haplotype Pattern Frequency Estimation

The following equation calculates the haplotype pattern frequencies for $C_{x,i}$ in a matrix $M$ or $M'$:

$$fr(M, C_{x,i}) = \frac{\sum_{j=1}^{2m} \prod_{y=1}^{n} F(C_{x,i}, j, y)}{2m} \tag{4.4}$$

where

$$F(C_{x,i}, j, y) = \begin{cases} 1, & \text{if } M[j,y] = C_{x,i}[y] \text{ and } C_{x,i}[y] \neq * \\ Pval(M, j, C_{x,i}[y]), & \text{if } M[j,y] = - \text{ and } C_{x,i}[y] \neq * \\ 0, & \text{if } M[j,y] \neq C_{x,i}[y] \text{ and } C_{x,i}[y] \neq * \end{cases} \tag{4.5}$$

The collaborated single genotype $C_{x,i}$ is a vector with length $n$ where each position represents an SNP and contains an allele for that SNP. Each position of this vector can have value 1, 0, or $*$ where 1 represents the major allele, 0 represents the minor allele

and $*$ indicates that this position is ignored. Equation 4.5 computes the frequencies by scanning the $C_{x,i}$ and ignores the computation for the positions $C_{x,i}[y] = *$, where $y \leq n$ . Hence, this computation allows the algorithm to compute frequency for haplotypes with non adjacent SNPs from matrix $M$ .

Frequency is computed by scanning each row in matrix $M$ at a time and matching the content of each position with the content of $C_{x,i}$. The matching of the contents of each position in $C_{x,i}$ with the matrix position $M[j, y]$ is computed by Equation 4.5. The function $F(C_{x,i}, j, y)$ returns a 1 if the value of the $yth$ position of $C_{x,i}$ is equal to the value of $jth$ row and $yth$ column of the matrix $M$ ; it returns 0 if the value do not match. The matched value of each row from Equation 4.5 are summed and divided by the total number of rows in matrix $M$ (see Equation 4.4). Hence, the function $fr(M, C_{x,i}$ returns the value between 0 and 1. The symbol " $-$ " in Equation 4.5 denotes the missing value in the j$th$ row and y$th$ column in matrix $M$. Function $Pval(M, j, C_{x,i}[y])$ returns an approximation value if there is a missing value in the matrices (see Section 4.3.3.2). These two equations apply to both $M$ and $M'$ to compute the frequencies of $C_{x,i}$ in these matrices.

#### 4.3.3.2 Handling Missing Data

In Section 2.1.3.4, we have stated that there can be missing data in haplotypes; hence in computing frequency for the haplotype pattern $C_{x,i}$, a technique is required to handle missing data in the given matrix. Previous missing-data-handling algorithms used various probabilistic methods. The most accurate of these used Bayesian method [54]. We have incorporated the concept of linkage disequilibrium (LD) for SNPs to compute approximate frequencies of an allele that is missing in the matrix $M$ . Among

the two prominent measures of LD we have discussed in Chapter 3, $r^2$ is the measure we have decided to use for our computation. The LD measure $r^2$ gives the correlation of alleles between a pair of SNPs [3]. Consider any two SNPs $A$ and $B$ with two alleles at each SNP $(a_1, a_2)$ and $(b_1,b_2)$, respectively. Let the observed frequency $P_A$ be the frequency of the first allele in SNP $A$ and $P_B$ be the frequency of the first allele in $B$. $P_{AB}$ is the observed frequency of haplotype that consists of the first alleles of $A$ and $B$. The disequilibrium measure $D$ is:

$$D = P_{AB} - P_A P_B \tag{4.6}$$

The $r^2$ between SNP $A$ and $B$ can be obtained by the following equation,

$$r^2 = \frac{D}{P_A \times (1 - P_A) \times P_B \times (1 - P_B)} \tag{4.7}$$

Let $L$ denote the LD matrix which is a $n\times$ $n$ matrix that stores all pairwise $r^2$ values of the $n$ SNPs from matrix $M$. We set a threshold that if the $r^2$ value between the pair is greater than 1/3 then the two SNPs are considered as linked [3]. Previous studies have reported that SNPs that are physically distant from each other seem to show weak linkage. Meanwhile, the decay of LD increases while the distance between SNP increases. An extensive amount of research has shown that SNPs can be linked with other SNPs that are up to 100kb apart [3, 43]. Given this information, the algorithm will consider a pair of SNPs as linked if their physical distance on the chromosome is within 100kb in addition to their $r^2$ being greater than 1/3.

When $M[j, y]$ contains a missing value " $-$ ", the function $Pval(M, j, C_{x,i}[y])$ finds a set of SNPs $Z$ from the matrix $L$ where each $z \in Z$ is strongly linked with the $yth$ SNP of matrix $M$, and uses that information to estimate the frequency of $C_{x,i}$ that

can occur in the missing position $M[j, y]$.

$$Pval(M, j, C_{x,i}[y]) = \frac{\sum_{z \in Z} fr(M, C_{x,i}[y], z)}{|Z|} \tag{4.8}$$

The construction of the set $Z$ returns a set of SNPs such that each $z \in Z$ is within 100kb with the *yth* SNP of matrix $M$ and the $r^2$ value between $z$ and $y$ is $> 1/3$. It is possible to have an empty set $Z$ for a set of loosely linked SNPs where all pairwise LD values in $L$ are $< 1/3$. In this case, the set $Z$ consists of all SNPs within 100kb of the *yth* SNP of matrix $M$.

The function $fr(M, C_{x,i}[y], z)$ in Equation 4.8 gives the average frequency of the allele $C_{k,i}[y]$ from matrix $M$ . The allele $C_{x,i}[y]$ can be a major or a minor allele. The idea of the function $fr(M, C_{x,i}[y], z)$ is to compute the average frequency of allele $C_{x,i}[y]$ from the SNPs that are in set $Z$ . This gives the approximate frequency of the allele $C_{x,i}[y]$ that can occur in a missing position of the matrix $M$ .

The fitness of a haplotype pattern $C_{x,i}[y]$ as defined in Equation 4.3 needs to be validated by statistical significance tests. However, since these tests are computationally expensive, only haplotype patterns whose fitness is above a threshold $t$ are statistically tested. This threshold will reduce running time by excluding patterns with low frequencies of occurrence. Haplotype patterns with low frequencies, e.g., <0.05, need a large sample size (i.e. a large number of case and control individuals) to obtain significant statistical results [40, 20, 50]. Hence, haplotype patterns with a frequency of $\geq t$ must have a frequency of $> 0.05$ in both case and control matrices.

#### 4.3.3.3 Solution Quality Tests

The next phase of the algorithm computes several statistical tests to measure the significance of the frequency calculated in Equation 4.3. These statistical tests do not affect the fitness but are used to quantify the quality of the solution.

**Haplotype Risk Ratio**

Haplotype Relative Risk (HRR) is a standard method for calculating the associated risk of a haplotype in a case-control study [16]. It defines the associated risk for each haplotypes for a disease carrier group or the cases. This test requires computation of occurrences or counts of the haplotype pattern in matrix $M$ and $M'$ instead of the frequencies. In Section 4.3.3, we have computed the case and control frequencies for $C_{x,i}$. To obtain occurrences of $C_{x,i}$ in case and control matrices, the numerator of Equation 4.4 is taken to compute HRR, which computes the count of $C_{x,i}$ in a matrix .

Let $a$ be the number of times haplotype $C_{x,i}$ occurred in case matrix $M$ and $b$ denote the number of occurrences in control matrix $M'$. The HRR for each $C_{x,i}$ is computed using a $2 \times 2$ contingency table as shown in Table 4.1 such that $HRR(C_{x,i}) = (a * d)/(b * c)$ . The value of HRR is considered to be significant if $HRR(C_{x,i}) \geq 1.5$ [34]. The HRR is calculated from a given case and control dataset; hence it is important to quantify the significance level of the computed HRR value of each haplotype pattern $C_{x,i}$ for this dataset. A Pearson's $\chi^2$ test is used to quantify the significance of the HRR value of the haplotype pattern $C_{x,i}$ for a given dataset and it has been suggested in the literature previously [9]. The $\chi^2$ test is only performed

| $C_{x,i}$ | Case Matrix M | Control Matrix M' |
|---|---|---|
| Count | a | b |
| Total Row − Count | c | d |

Table 4.1: Contingency Table for Computing Haplotype Relative Risk for $C_{x,i}$.

on those haplotypes whose HRR $\geq$ *1.5* to reduce computation time by ignoring haplotypes with low or negligible relative risks. False positive results are possible after getting significant $\chi^2$ values. To avoid false negative results, the $p$-value of a $C_{x,i}$ is considered to be significant if $p$-value is $\leq 0.001$ with a $\chi^2 \geq$ *10.83* [35].

**Permutation Test**

As stated above, the $\chi^2$ test is used to quantify the significance of a haplotype pattern's HRR value for the given dataset. The HRR value may show strong significance in the given dataset, which can be interpreted as significant by chance or as Type 1 error. Hence, we need to determine the global significance of each haplotype pattern $C_{x,i}$. The Permutation test detects Type 1 error but is computationally expensive [66]; hence, it is only computed on those haplotype patterns where the computed HRR value is $> 1.5$ and the associated $\chi^2 \geq$ *10.83* . This can be accurately validated using a permutation test as follows:

1. Let $V$ be the value for $\chi^2$ for the HRR of the haplotype pattern $C_{x,i}$ from matrices $M$ and $M'$. This value $V$ represents the $\chi^2$ value computed to evaluate the significance of HRR computed from the given $M$ and $M'$ matrices. Set *count* $= 0$.

2. Randomly reshuffle the case and control labels of each haplotype pair in matrix $M$ and $M'$ to obtain new case-control matrices.

3. Compute $\chi^2$ value $V'$ for the haplotype $C_{x,i}$ from the new matrices obtained in Step 2.

4. If $V' \geq V$ *increment* count .

5. Repeat Steps 2-4 10,000 times.

The empirical $p$-value can be obtained by dividing *count* by 10,000. The $p$-value is considered to be significant if it is $\leq 0.0005$ .

#### 4.3.3.4 Niching

Maintaining the diversity of members in a species helps prevent a population from reaching premature convergence to a local optimum. Niching is a method that maintains diversity and prohibits different members of that species from crowding into the same area of the solution search space [32]. One common niching technique is *fitness sharing*, in which members of a species that are close to each other in solution search space have to share their fitness with each other.

Two members are considered close if the distance between their associated genotypes is within a certain threshold. In our algorithm, members are considered close if the Hamming distance between their genotypes is $\leq 3$. For example, if two members within a species have chromosomes 10101*01*0 and 10001*01*0 so that the Hamming distance between them is 1, a 15 % penalty will be applied to one of them, decided randomly. There can be a set of members that are within the threshold of

Hamming distance with another member, in which case the penalty applies to all the members in that belonging to that set. The rate of penalty is an important issue in maintaining diversity as well as promoting a population to evolve more fit solutions. A high penalty rate might misdirect the evolutionary search. We therefore decided a relatively moderate penalty rate of 15% which will be reduced from the current fitness of a member $p_{x,i}$. This penalty reduces the probability of member $p_{x,i}$ being selected to form the next generation.

### 4.3.4 Genetic Operators

In our CCGA scheme, the one-point crossover operator is applied to a pair of parents in a species $S_x$ to produce an offspring. As stated in Section 4.3.2, each member in a species is a vector with fixed length $l$ where at each position there can be 0,1 or *. A random point $r$ is drawn from the range 1 to $l$, where $l$ is length of the member chromosome for that species (see Figure 4.10($a$)). The offspring is produced by copying positions 0 to *r-1* from the first parent and positions $r$ to $l$ from the second parent. The second genetic operator is the mutation operator. We have adapted the one point bit-flip mutation operator to produce an offspring (see Figure 4.10($b$)). The mutation operator operates on a parent from a species $S_x$ which is governed by the mutation rate. A position in the parent's vector is chosen randomly and that position is flipped by the following rules:

- if there is a 1, the bit is flipped to 0
- if there is a 0, the bit is flipped to $*$
- and if there is a $*$, the bit is flipped to 1.

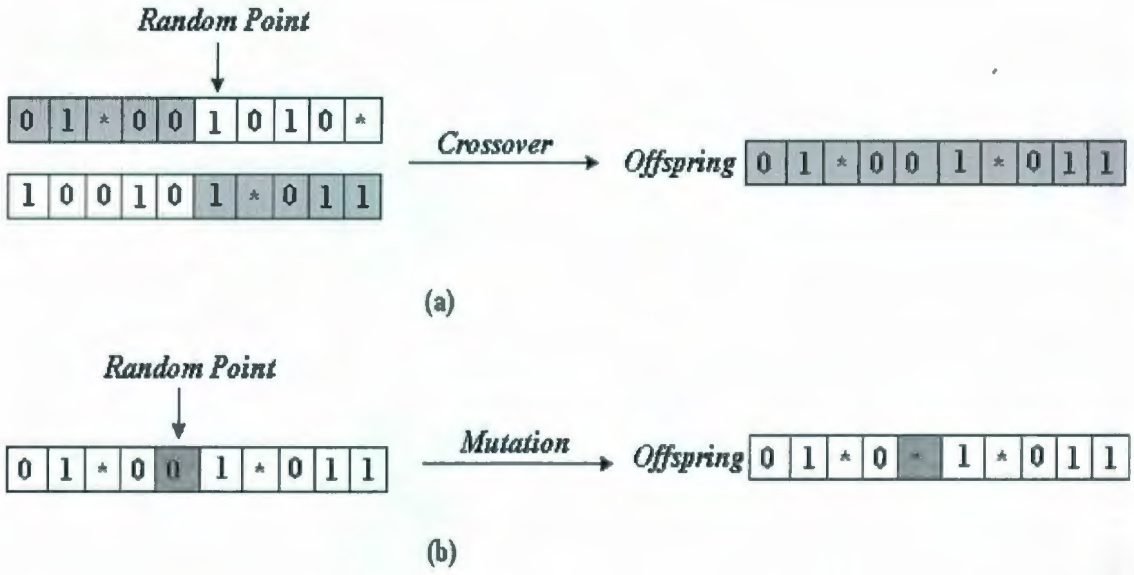


Fig. 4.10: Operation of Crossover and Mutation Operators. (a) Crossover operator on two parents. (b) Mutation operator on one member.

### 4.3.5 Selection

We have used random selection to select parents for the genetic operator application. During crossover, a pair of parents is randomly selected from the same species and for mutation one parent is randomly selected from a species.

The proposed CCGA is steady-state, such that parents and offspring can compete with each other in the survival selection process. This algorithm does not maintain replacement rate; hence, any parent member can be replaced by a better offspring. A wide variety of techniques exist that implement selection and each of these selection techniques has their own pros and cons. In our selection technique, we have performed a pairwise competition between a parent and an offspring such that the more fitter of the two is selected and kept for the next generation.

| *Parent* | *Offspring* | *f(Parent)* | *f(Offspring)* | *Selected* |
|---|---|---|---|---|
| a | $c'$ | 0.16 | 0.99 | $c'$ |
| b | $a'$ | 0.63 | 0.12 | b |
| c | $e'$ | 0.12 | 0.19 | $e'$ |
| d | $b'$ | 0.49 | 0.96 | $b'$ |
| e | $d'$ | 0.96 | 0.61 | e |

Table 4.2: Example of the Selection Technique Used in the HPD CCGA.

Table 4.2 gives an example of the selection mechanism that we have developed in our CCGA for a species with 5 members. The pairing of parents and offspring is random. However, each pair only appears once. In this way, the fittest member is selected only once, which gives other members an equal probability to be selected for the next generation. This selection design of pairing parents and offspring randomly maintains the properties of the steady-state population.

# Chapter 5

# Case Study Results

The CCGA algorithm for the haplotype pattern detection problem was implemented in a Java software package using JDK 1.5. The implemented software takes haplotypes of case and control data and applies the algorithm to detect haplotype patterns that are susceptible to a disease. Since computational time is a crucial factor for any algorithm we use numerical representation for the given case-control matrices instead of string representation because string computation is much more expensive. The implemented algorithm converts the string haplotype data into a binary matrix where 1 represents the major allele and 0 represents the minor allele for each SNP site in the case-control matrices.

The algorithm described in the previous chapter needs to be tested against published datasets in order to measure the effectiveness of this algorithm against others proposed in the literature. In Section 5.1, the data sets that were used in our experiments are described. In Section 5.2, the parameter setup for each experiment is given. CCGA performance is discussed in Section 5.3 and the haplotype patterns captured

by the CCGA from the datasets are discussed in Section 5.4. Finally, in Section 5.5, the limitations and the future work of the proposed CCGA are outlined.

## 5.1 Dataset Descriptions

The algorithm was applied to three different cohorts for two complex diseases. The original SNP datasets that we obtained consisted of genotype data. As stated in Chapter 3, our algorithm is designed for haplotypes; hence we applied a well known phasing algorithm implemented in the application PHASE v2.0 to obtain the haplotypes for all three cohort genotype data. We have chosen PHASE v2.0 because the performance of PHASE v2.0 is best among all the existing phasing algorithms. The most accurate haplotype pair for each individual in a cohort obtained from the PHASE v2.0 application was converted into numerical format as described above.

The first two cohorts were genotyped from two Canadian populations for the disease Ankylosing Spondylitis(AS). AS is the most common cause of inflammatory arthritis and the genetic behavior of this disease is yet to be analyzed [8]. The most significant gene that is associated with AS is the HLA-B27 gene located on chromosome 6. Published research suggests that HLA-B27 operates by combining with other genes. Among these other genes, the IL1 gene cluster has shown susceptibility to AS.

The datasets for the AS cohorts were obtained from Maksymowych *et al.* [37], where the authors performed a haplotype association analysis on three Canadian cohorts from Alberta, Newfoundland, and Toronto. The Toronto dataset is not ethnically matched and our algorithm does not handle population admixture. Hence, we have studied two ethnically matched cohorts which were genotyped from Alberta

and Newfoundland populations. The genotypic region spans a 360kb that includes the IL1 gene cluster located on human chromosome 2. Initially, 38 SNPs were genotyped which include the IL1 gene cluster consisting of the IL1A, IL1B, IL1F7, IL1F9, IL1F6, IL1F8, IL1F5,IL1F10, and IL1RN genes. A core set of 20 SNPs were kept for haplotype pattern detection and 18 SNPs were excluded from the analysis by HWE deviation and tag SNP criteria (see Table 5.1). In both cohorts, the individuals that were genotyped were unrelated to each other, i.e., no familial relationship exists between the individuals. The Alberta cohort includes ethnically matched 200 white healthy controls and 200 AS patients. The obtained haplotypes from PHASE v2.0 had 1.14% and 0.75% missing data in the case and control data, respectively. The Newfoundland cohort is relatively small with 150 white healthy controls and 112 AS patients. The haplotypes in this cohort contain 0.12% and 0.62% missing data in the case and controls, respectively.

Another disease that was taken into consideration for our study is Schizophrenia. About 1% of the population is affected by this complex disease [49]. This disease affects an individual by hereditary (inherited from family members) or by other environmental and biological causes (i.e. infections, drug side effects). Different research suggests that different areas of the human genome are associated with Schizophrenia and further investigation is required to pinpoint genomic regions for this disease. Nevertheless, the Netrin G1 gene has been suggested as one important region that shows susceptibility with Schizophrenia [2].

The third cohort in our experiment was taken from a previous study by Fukasawa *et al.* [18] that investigated Schizophrenia in the Japanese population. This cohort is the smallest among the three. The genotypic region consists of 10 SNPs that span

| SNPs | Major/Minor Allele | Chromosome Position | Gene |
|---|---|---|---|
| 1. rs2856836 | T/C | 113627229 | IL1A |
| 2. rs3783550 | A/C | 113628031 | IL1A |
| 3. rs3783547 | T/C | 113628485 | IL1A |
| 4. rs3783543 | C/T | 113631797 | IL1A |
| 5. rs17561 | G/T | 113632369 | IL1A |
| 6. rs3783526 | G/A | 113636953 | IL1A |
| 7. rs1800794 | C/T | 113638419 | IL1A |
| 8. rs1143643 | G/A | 113683448 | IL1B |
| 9. rs1143634 | C/T | 113685536 | IL1B |
| 10. rs1143630 | C/A | 113686801 | IL1B |
| 11. rs3917356 | G/A | 113687509 | IL1B |
| 12. rs3917354 | A/C | 113688041 | IL1B |
| 13. rs1143627 | T/C | 113689533 | IL1B |
| 14. rs3811047 | G/A | 113766556 | IL1F7 |
| 15. rs2723187 | C/T | 113770415 | IL1F7 |
| 16. rs895497 | C/T | 113858721 | IL1F6 |
| 17. rs1900287 | A/G | 113892711 | IL1F8 |
| 18. rs3811058 | T/C | 113927091 | IL1F10 |
| 19. rs419598 | T/C | 113982349 | IL1RN |
| 20. rs315951 | G/C | 113985729 | IL1RN |

Table 5.1: The SNPs in the IL1 Gene Cluster.

| SNPs | Major/Minor Allele | Chromosome Position | Gene |
|---|---|---|---|
| 1. rs4481881 | T/C | 105861098 | Netrin G1 |
| 2. rs4307594 | T/C | 105867847 | Netrin G1 |
| 3. rs3924253 | A/G | 105895683 | Netrin G1 |
| 4. rs4132604 | G/T | 106017575 | Netrin G1 |
| 5. rs3762369 | C/T | 106112333 | Netrin G1 |
| 6. rs894904 | T/C | 106122620 | Netrin G1 |
| 7. rs2218404 | G/T | 106127339 | Netrin G1 |
| 8. rs1373336 | C/T | 106152557 | Netrin G1 |
| 9. rs1444042 | A/G | 106164632 | Netrin G1 |
| 10. rs96501 | T/C | 106193594 | Netrin G1 |

Table 5.2: The SNPs in the Netrin G1 Gene.

a 40kb region located on human chromosome 1p13.3 (see Table 5.1). This region includes the Netrin G1 gene which is also known as Laminet 1. The cohort consists of 180 healthy controls and an equal number of schizophrenia patients. The haplotypes obtained from PHASE v2.0 provided complete haplotype pairs for each individual i.e. there is no missing or ambiguous data.

## 5.2 Experiment Setup

The experiments on the three cohorts were carried out using the CCGA parameters given in Table 5.2. The optimal values for these parameters are not known for this problem but general knowledge adapted from different genetic algorithm applications was used to maintain a balance so that the algorithm's evolution is not so disruptive that it is effectively performing random search. The crossover rate was set to 100% because one-point crossover is not as disruptive as uniform crossover [71]. To exploit the fitness landscape with adequate chromosomal swapping, this rate of crossover is favorable. The mutation rate was set to 5%. It is possible that the selected parameter values in Table 5.2 do not give the optimal performance. Since the objective of this thesis is to demonstrate that the proposed CCGA algorithm can solve the HPD problem, not to design the most efficient CCGA algorithm, we only conduct experiments using this set of parameters. In our future work, we will investigate CCGA performance relative to other parameters sets for this problem.

Each cohort was split into two species where each species contains an equal number of SNPs. The AL and NF cohorts contain 10 SNPs in each species. The population size of each species was set to 250 in the AL and NF cohorts. According to Equation

4.1, the Alberta AS cohort consists of $\leq 3^{20+1} \times 200$ or $\leq$ 2,092,070,640,600 possible haplotype patterns and the Newfoundland AS cohort includes $\leq 3^{20+1} \times 150$ or $\leq$ 1,569,052,980,450 possible haplotype patterns. To search the large spaces in these two cohorts, a population size of 250 provides sufficient diversity for the population to evolve. Hence, the CCGA will examine $(2 \times 250 \times 2) \times 1000 = 1{,}000{,}000$ haplotype patterns to find significant solutions in each run on the two cohorts. The Schizophrenia cohort contains 5 SNPs in each species because of its small number of SNPs. The entire search space of this cohort consists of $\leq 3^{10+1} \times 180$ or $\leq$ 31,886,460 possible haplotypes. The population size for the Schizophrenia cohort was set to 25 because the dataset is smaller than the AS cohorts. The CCGA will evaluate $(2 \times 25 \times 2) \times 1000 = 100{,}000$ haplotype patterns in each run. Note that in each cohort, the CCGA only samples a small portion of the search space to find significant haplotype patterns.

The fitness threshold was set to $t \geq 0.10$. This is the smallest fitness value of a haplotype pattern that enables performance of statistical significance tests. The niching penalty was set to 15%, i.e., any member that is close (Hamming distance is $< 3$) to another member will be penalized by reducing the current fitness by 15%.

## 5.3 Performance Evaluation

As stated in Chapter 4, CCGA evolves multiple populations; hence, the evidence of evolution in individual species validates the behavior of a CCGA algorithm performance. Figure 5.1 shows the evolution of different species in a single CCGA run on the three cohorts. The plot diagrams of each cohort were generated by computing the

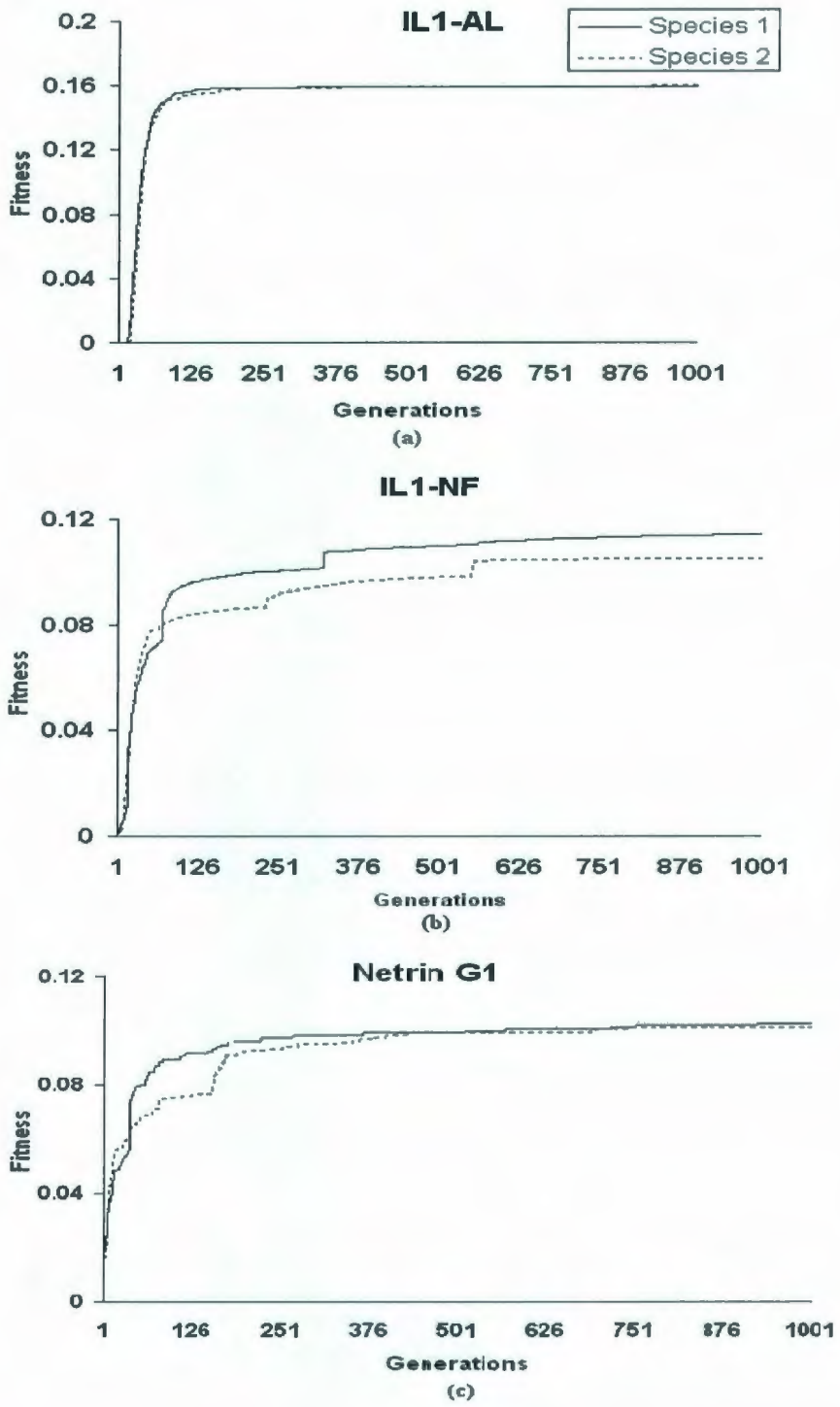


Fig. 5.1: Evolution of Multiple Species in a Typical CCGA Run. The average fitness for each species was plotted for each generation relative to a single CCGA run.

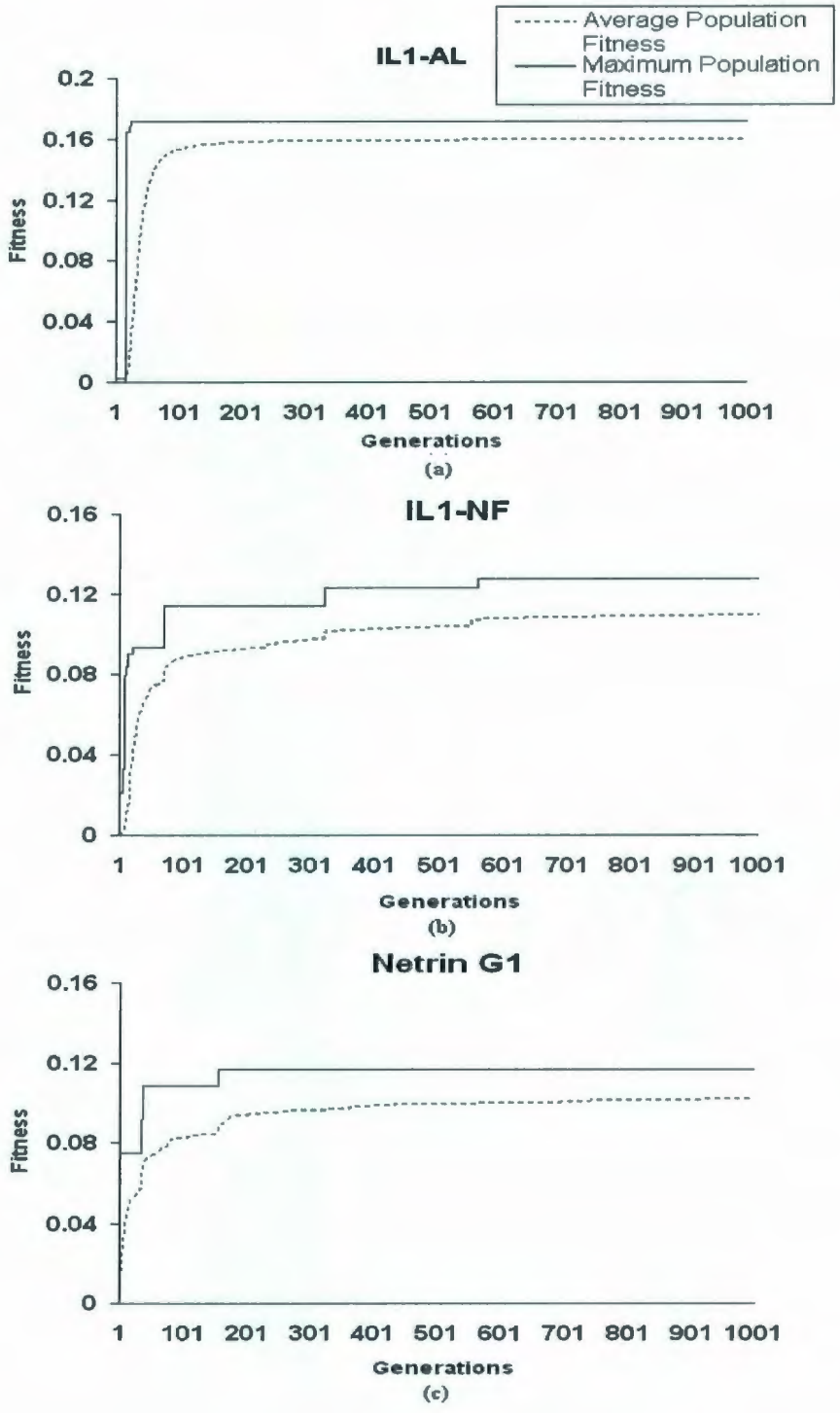


Fig. 5.2: Average Fitness for a Single CCGA Run (AL, NF, and Japanese Cohorts).

| Parameters | IL1-AL | IL1-NF | Netrin G1 |
|---|---|---|---|
| Number of Generations | 1000 | 1000 | 1000 |
| Crossover Rate | 100% | 100% | 100% |
| Mutation Rate | 5% | 5% | 5% |
| Number of Species | 2 | 2 | 2 |
| Number of SNPs in Each Species | 10 | 10 | 5 |
| Population Size in Each Species | 250 | 250 | 25 |
| Fitness Threshold *(t)* | 0.10 | 0.10 | 0.10 |
| Number of Runs | 100 | 100 | 100 |
| Niching Penalty | 15% | 15% | 15% |
| Number of Permutation Test | 10000 | 10000 | 10000 |

Table 5.3: CCGA Parameters and Their Values.

average population fitness of each species in each generation from a single run. Figure 5.1(a) shows that the fitness of both species converges around generation 100 on the AL cohort. The population fitness improvement was steady with a high increase in the fitness value before generation 100. Both species evolve simultaneously from the beginning and some fitness differences can be observed between the two species until the population fitness converges. Similarly, the NF cohort shows a fast increase of average population fitness values in both species until generation 110 (see Figure 5.1(b)). After that, the fitness improvements are small.

The Schizophrenia cohort is the smallest cohort among the three and the experimental setup was different. The simultaneous evolution of each species in Figure 5.1(c) shows similar behavior where the two species' average population fitness im-

proves very fast until generation 150, when it converges with a small subsequent increase in the fitness value.

The performance of individual species has been discussed based on a single run and it is important to investigate the combined performance of the species of the CCGA in different cohorts. In Fig 5.2 the combined average population fitness and the average maximum population fitness of both species are plotted for the three cohorts from a single run. The typical run of the proposed CCGA in the AL cohort showed that the species converges after generation 80, and both the average population fitness and the average maximum population fitness converge. In contrast, the NF cohort fitness curve is not as smooth as the AL cohort curve; the NF cohort shows that an increase on maximum population fitness also affects the average population fitness. The combined fitness in the Schizophrenia cohort shows rapid evolution until generation 180 and after that, the average maximum fitness completely converged such that the average fitness curve shows small increases but the maximum fitness remains unchanged.

The algorithm performance was different in terms of computation time. The computation varies in each run because of the execution of the permutation test. In a single run for any three cohorts, if the CCGA finds more significant haplotypes, then it takes more time to compute because of the permutation tests. The approximate time for a single CCGA run in the AL and NF cohorts was approximately 16 minutes and the Schizophrenia cohort running time was approximately 7 minutes on a Pentium 1.73Ghz machine. This computation time is significantly smaller than that for classical statistical algorithms. One classical statistical algorithm that we have studied for time comparison is WHAP. The application WHAP applies an omnibus

statistical test in each window to quantify the significance by a $p$-value [37]. We have tested WHAP on the IL1 cohort with a 3 SNP sliding window analysis. Each window takes about 1.5 hours to compute the p-value after 10,000 permutations on a pentium 1.73GHz machine. Since the IL1 data consists of 20 SNPs, which give us 18 distinct windows (each window with 3 SNPs), it will take approximately 27 hours to compute all 18 windows. The execution time of our algorithm for 1000 generations on the IL1-AL and IL1-NF cohorts takes about 20 minutes for each. It shows our method has the potential to be applied to moderately large sized datasets.

The performance of a genetic algorithm should not be evaluated by its single run performance. The performance of the CCGA from a single run does not demonstrate the consistency because the result can be obtained by chance. It is wise to evaluate the performance of the CCGA by executing multiple runs on the same datasets. We have executed the proposed algorithm 100 times on each of the three cohorts. The average population fitness and the average maximum population fitness of each run was computed to observe the consistency.

In the AL cohort we can see a very consistent performance of the algorithm. In Figure 5.3(a), the fitness plot of 100 runs demonstrates that these results are not obtained by chance or random search; instead it shows consistent performance of the evolutionary process of the proposed CCGA. The minimum value of the average population fitness in the AL cohort among the 100 runs is 0.063, which is an outlier as shown in Figure 5.4. This outlier does not represent the overall performance of the CCGA in the AL cohort because the mean of the average population fitness in 95% of the runs lies between 0.116 and 0.127. This consistency is also observed in the average maximum population fitness, where the mean of the average maximum

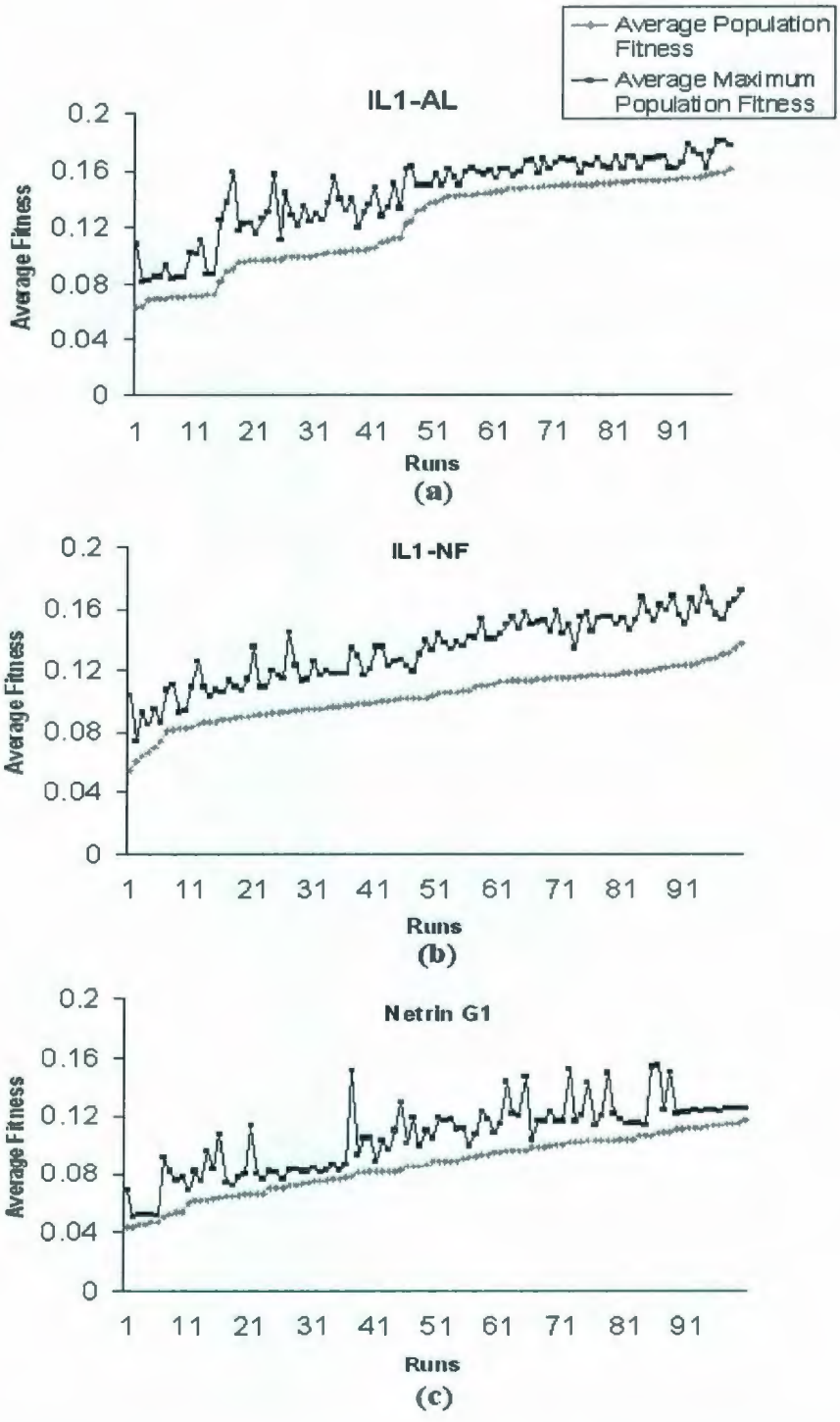


Fig. 5.3: Average Fitness for 100 Runs (AL, NF, and Japanese Cohorts).

population fitness in 95% of the runs lies between 0.138 to 0.150 (see Figure 5.3 and Table 5.4).

The NF cohort also shows a similar behavior where the average population fitness in most of the runs lies between 0.099 and 0.107. The minimum average population fitness among this 100 runs of the NF cohort (0.05) is also an outlier which does not represent the average performance of the algorithm (see Figure 5.3(b) and Table 5.4). The average maximum population fitness also shows consistent performance, where 95% of the runs shows the mean of the average maximum population fitness value between 0.130 and 0.137.

The schizophrenia cohort shows less consistent performance, where 95% of the runs shows average population fitness mean is between 0.081 and 0.091. There exist few outliers in the average population fitness plot from the 100 runs. The average maximum population fitness in the Schizophrenia cohort shows consistent performance, where 95% of the runs showed the average maximum population fitness mean is between 0.10 and 0.11 (Figure 5.3(c) and see Table 5.4).

We have observed the consistent performance of the CCGA algorithm from the 100 runs. The algorithm always maintained a significant difference between its average population fitness and average maximum population fitness (see Figure 5.3). The box plot in Figure 5.4 shows the mean of average population fitness and the average maximum population fitness is different in all three cohorts. It is important to quantify the relationship between the average population fitness and the average maximum population fitness by performing statistical tests. This statistical significance of the difference between the two fitnesses (average population fitness and average maximum fitness) obtained from the 100 runs will establish consistent performance

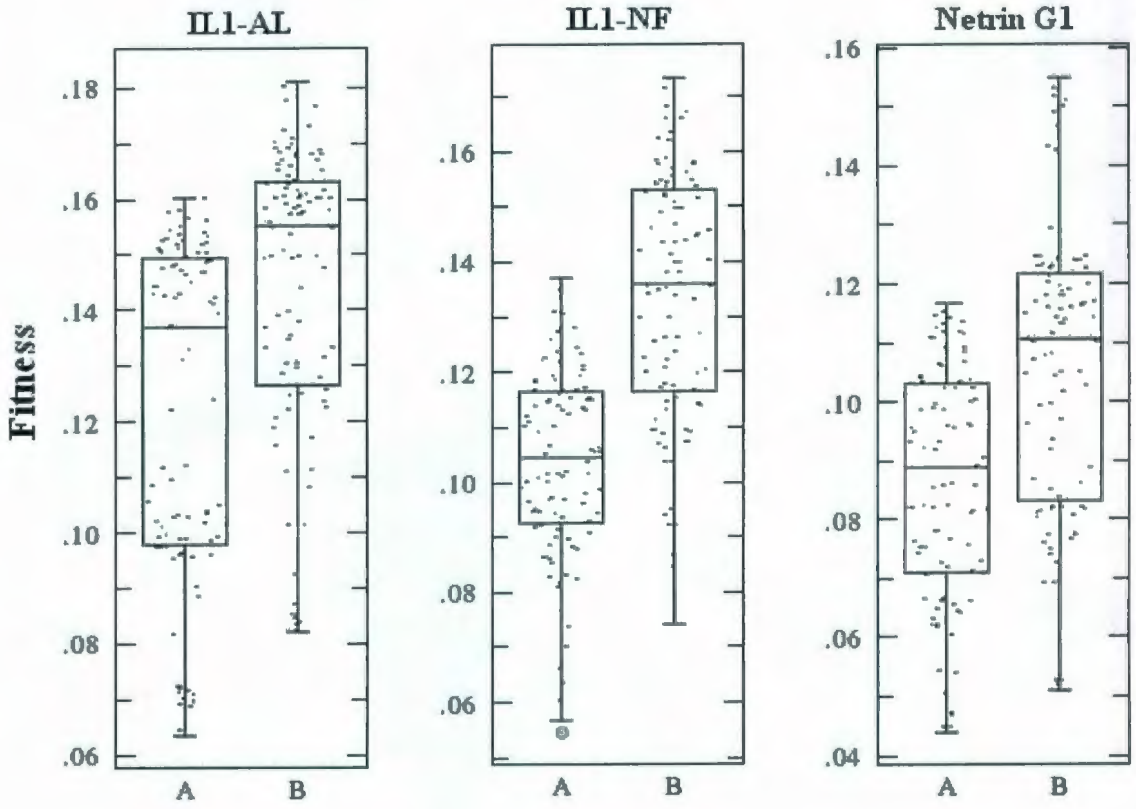


Fig. 5.4: Box Plot of Fitness for 100 Runs (Al, NF, and Japanese Cohorts). In each box, the points on column A represent average fitness values of each run and points on column B represent average maximum fitness of each run.

of the proposed CCGA algorithm for the HPD problem. The 100 run results have been tested using the t-test to improve the confidence of the result.

A t-test computes the probability that two datasets are different. The null hypothesis of this test considers that there is no difference between the mean values of the two data series and the alternate hypothesis is otherwise. A $p$-value with the significance level $< 0.05$ implies that the mean difference between the two data series is not due to chance; hence it is sufficient to reject the null hypothesis. We have applied the t-test on the two data series where one set of data points is the average

**Average Fitness**

| Cohort | Runs | Average | Min | Max | Std | Mean(95%$CI$) |
|---|---|---|---|---|---|---|
| IL1-AL | 100 | 0.121 | 0.063 | 0.160 | 0.030 | 0.122(0.116 – 0.127) |
| IL1-NF | 100 | 0.103 | 0.05 | 0.13 | 0.016 | 0.104(0.099 – 0.107) |
| Netrin G1 | 100 | 0.085 | 0.043 | 0.116 | 0.019 | 0.085(0.081 – 0.091) |

**Average Maximum Fitness**

| Cohort | Runs | Average | Min | Max | Std | Mean(95%$CI$) |
|---|---|---|---|---|---|---|
| IL1-AL | 100 | 0.143 | 0.082 | 0.180 | 0.027 | 0.144(0.138 – 0.150) |
| IL1-NF | 100 | 0.133 | 0.074 | 0.173 | 0.022 | 0.134(0.130 – 0.137) |
| Netrin G1 | 100 | 0.104 | 0.051 | 0.154 | 0.024 | 0.105(0.100 – 0.110) |

Table 5.4: Distribution Characteristics of Average Population and Average Maximum Population Fitness for 100 Runs (AL, NF and Japanese Cohorts).

population fitness and the other set is the average maximum population fitness obtained from the 100 CGGA runs. Table 5.5 shows that the difference between these two fitness values for each cohort is significant ($p < 0.0001$). These statistical tests justify the conclusion that evolutionary force directs the search performance of the proposed CCGA scheme.

## 5.4 SNP Cohort Results

In the previous section, we evaluated the performance of the proposed CCGA algorithm in terms of evolutionary behavior of the algorithm. The performance of the algorithm also needs to be analyzed based on the detected haplotype patterns from

| Cohort | Run | stdv | t | p-value |
|---|---|---|---|---|
| IL1-AL | 100 | 0.028 | 5.32 | <0.0001 |
| IL1-NF | 100 | 0.019 | 10.8 | <0.0001 |
| Netrin G1 | 100 | 0.022 | 6.05 | <0.0001 |

Table 5.5: Significance Tests of Average Population Fitness and Average Maximum Population Fitness (AL, NF and Japanese Cohorts).

the three cohorts using the proposed CCGA and how these results compare with previous results obtained using classical statistical techniques. In Section 5.4.1, the two Ankylosing Spondilitis (AS) cohort results will be discussed and in Section 5.4.2, the Schizophrenia cohort results are discussed.

### 5.4.1 Ankylosing Spondylitis (AS) Data

In this section we will discuss the results obtained for the two AS cohorts after running the CCGA 100 times, and compare these results with the published results by Maksymowych *et al.* [37]. In that work, the authors conducted their analysis in two phases. In the first phase, they performed single window analysis, in which they obtained 4 SNPs (rs3783550, rs3783543, rs3783526 and rs1143627) with significant association with AS in the AL cohort. In the second phase, the authors performed omnibus statistics in each 3-window haplotypes, and the final $p$-value is obtained by permuting the data 10,000 times. They reported that several haplotype windows in the IL1A, IL1B and IL1F7 genes show significant susceptibility to AS in the AL cohort. The most significant haplotypes in the IL1A and IL1B genes were obtained from the SNPs rs3783543, rs17561, rs3783536, rs1800794, rs1143643, rs1143634, rs1143630,

rs3917356, rs3917354,and rs1143627. The authors did not find any susceptible SNPs or haplotypes in the NF cohort.

The same region (IL1 gene cluster) has also been analyzed by Timms *et al.* [59] in a British parent-case study (i.e. parents are controls and affected siblings are cases). The authors have performed both single window and 2-window analyses. They reported that strong association was found in IL1B, IL1F8 and IL1F10. The authors also point out the weak association in the IL1F7 gene but have not found any weak or strong association in the IL1A gene.

In our results, 53 significant haplotype patterns were identified in the AL cohort with a global $p < 0.0005$ and HRR $\geq 1.5$ after permuting the case-control data 10,000 times. The haplotypes that we found from the AL cohort contain all major alleles from SNPs rs3783550, rs3783543, rs3783526, rs1143630, rs3917354, rs1143627,rs2723187,and rs3811058 (see Figure 5.5 and 5.6). These SNPs include the genes IL1A, IL1B, IL1F7, and IL1F10. In our results, the SNPs rs3783550, rs3783543, rs3783526 and rs1143627 were also identified by the Maksymowych *et al.* single window analysis. The significant haplotype patterns obtained by the CCGA from the SNPs rs3783550, rs3783543, rs3783526, rs1143630, rs3917354, rs1143627, and rs3811058 that includes IL1A, IL1B and IL1F10 genes are in strong accordance with two previous independent studies [37, 59].

The haplotype patterns that we have found show susceptibility of the IL1F10 gene with AS, which contradicts Maksymowych *et al.* but is in strong agrement with Timms *et al.* The SNP rs2723187 with underlying gene IL1F7 shows relatively weak association in our results and very few haplotypes detected by our CCGA include the SNP rs2723187. The weak association of this SNP agrees with that reported

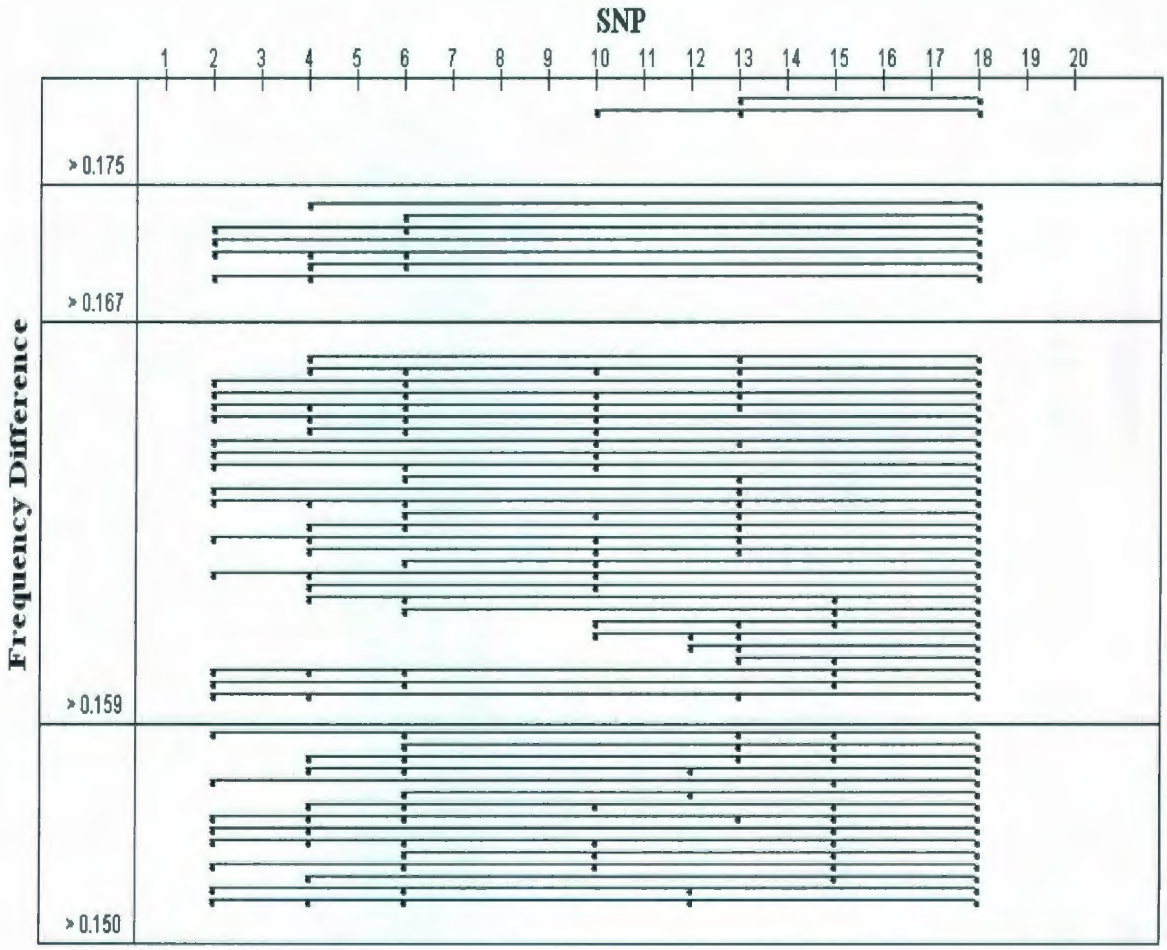


Fig. 5.5: Haplotype Patterns Captured from 100 Runs (AL cohort).

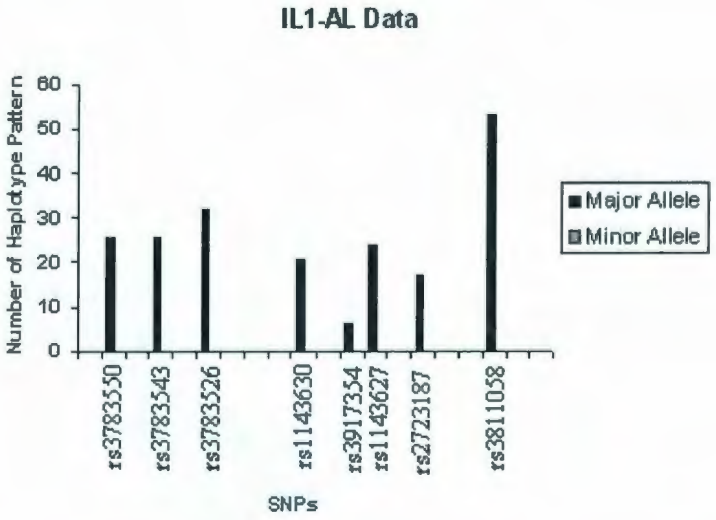


Fig. 5.6: Number of Haplotype Patterns that are Obtained from Each SNP (AL cohort).

by Timms *et al.* The most significant haplotype $TT$ was captured from the SNPs rs1143627 and rs3811058 with a frequency difference (in case and control matrix) of 0.18 and a HRR is 2.19 with a global $p < 0.0001$. Most of the haplotype patterns we obtained from the AL cohort contain a suffix of these two alleles. The haplotype $ACGCTT$ is the longest haplotype captured in the AL cohort with a frequency difference of 0.16 and the HRR is 1.96 with a global $p < 0.0001$ that is obtained from SNPs rs3783550, rs3783543, rs3783526, rs1143630, rs1143627 and rs3811058 (see Figure 5.5).

The results that we have obtained from the NF cohort using the CCGA are not significant in terms of the number of haplotype patterns detected by the algorithm. Only two haplotypes were detected with smaller frequency differences between case and controls (see Figure 5.7 and 5.8). The haplotypes with significant association with AS obtained from the NF cohort include genes IL1A, IL1B and IL1RN. The

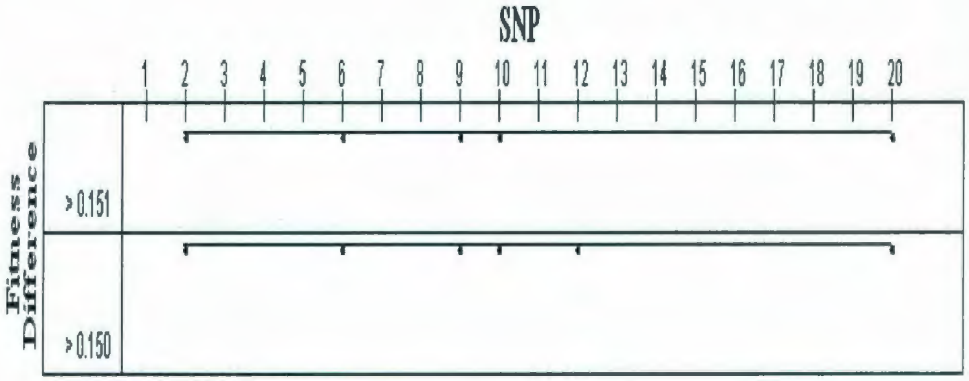


Fig. 5.7: Haplotype Patterns Captured from 100 Runs (NF cohort).

most significant haplotype observed in this cohort is *AGCCTG* and the HRR of this haplotype is 2.12 with a $p$-value of 0.0002 which is obtained from SNPs rs3783550, rs3783526, rs1143634, rs1143630, and rs315951 (see Figure 5.7).

### 5.4.2 Schizophrenia Data

Two analyses have been done in the Netrin G1 gene region that is located on chromosome 1. Fukasawa *et al.* [18] conducted a case-control cohort analysis from the Japanese population that included genotypes for 10 SNPs in this Netrin G1 region. The authors performed single window, 2-window and 3-window analyses to evaluate SNPs and their underlying haplotypes for possible susceptibility to schizophrenia.

In their single window analysis, they have found significant association ($p < 0.05$) in SNP rs1373336. The 2-window and 3-window analyses showed significant association with haplotypes from SNPs rs894904, rs2218404, rs1373336, and rs1444042. The authors concluded that rs1373336 was the most significant SNP which has been detected by the three different analyses.

In another independent study [2], the authors performed a family based analysis

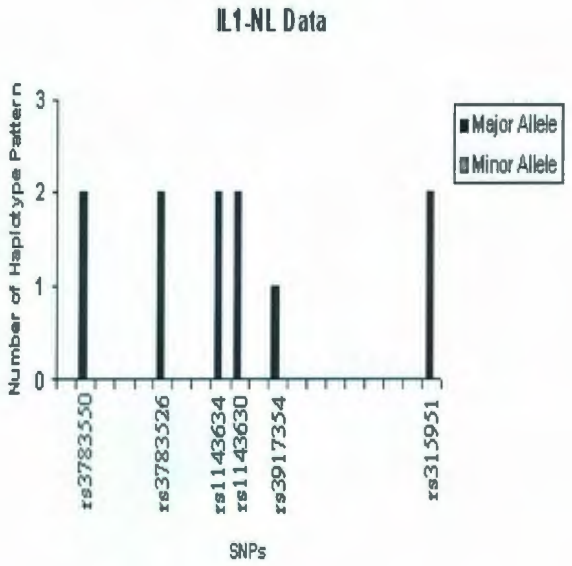


Fig. 5.8: Number of Haplotype Patterns that are Obtained from Each SNP (NF cohort).

on a broader chromosomal region that also includes the 10 SNPs in the Netrin G1 gene region. The authors reported strong association of SNPs rs4307594, rs3924253, rs1373336 and rs96501 in their single window results. In their 3-window analysis, the SNPs rs4307594, rs3924253, rs4132604, rs2218404, rs1373336, and rs1444042 showed susceptibility to Schizophrenia. The results in both studies detected the susceptibility of rs1373336 to Schizophrenia.

In our results, we have found 8 haplotype patterns consisting of SNPs rs4481881, rs4307594, rs3924253, rs4132604, rs1373336, rs1444042 and rs96501 (see Figure 5.9 and 5.10). Our results also show that rs1373336 is the most significant SNP because all the haplotype patterns detected by the CCGA from the Schizophrenia cohort contain the major allele from SNP rs1373336. The haplotype patterns that were captured by the CCGA contain major alleles in all SNPs except rs4132604, where the

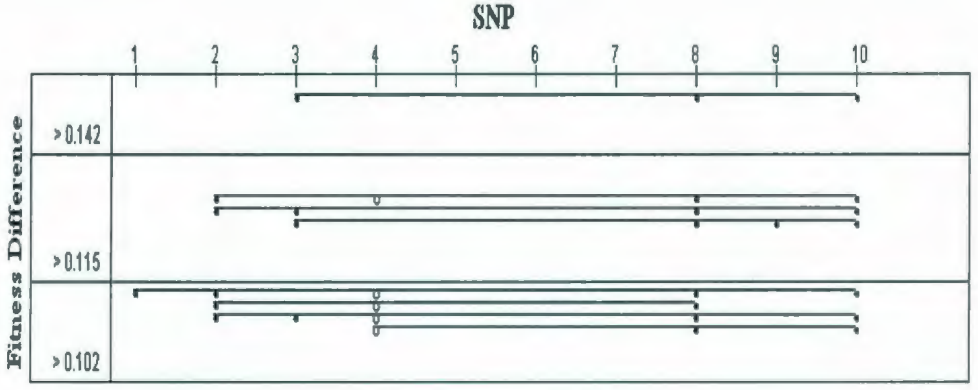


Fig. 5.9: Haplotype Patterns Captured from 100 runs (Japanese cohort).

rare allele was included in the haplotype patterns. The most significant haplotype captured by our algorithm is $ACT$ from SNP rs3924253, rs1373336, and rs96501, such that the HRR is 2.34 with $p < 0.0001$.

## 5.5 Algorithm Limitations and Future Work

In light of our results above, there are some significant issues that our algorithm design did not consider. One such issue was the population stratification bias. Recent studies have suggested that there exist significant differences in different population genetic maps and genomes. These studies also have shown different populations contain different haplotype structure [19, 26, 58]. The reasons for these differences include environmental effects, diseases etc. It is important that all population-based genetic research acknowledge this stratification bias. Our proposed CCGA algorithm is designed to detect SNPs and the underlying haplotypes from a case-control cohort that is ethnically matched. In other words, the individuals in the cases and the controls must have the same ethnic background. A cohort with mixed ethnicity

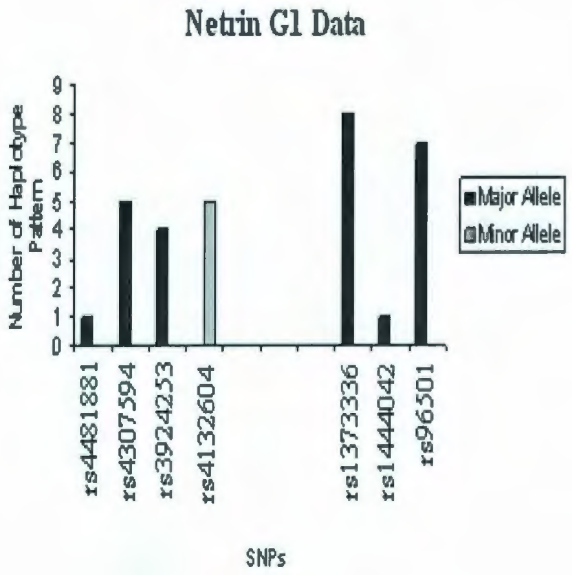


Fig. 5.10: Number of Haplotype Patterns that are Obtained from Each SNP (Japanese cohort).

will disrupt the accuracy of the algorithm because mixed ethnicity will introduce a stratification bias which this algorithm does not consider while computing haplotype frequencies. Hence, using a mixed population in a cohort might produce false positive results.

Statistical tests often produce false positive results and there exist various techniques to reduce these false positive results. One prominent method is called multiple test correction. Multiple correction tests multiply obtained $p$-values from a series of tests by the number of tests performed. Hence, for a large number of tests, the multiple correction test is not applicable because to pass multiple correction test the $p$-values must be really small [42, 41]. In our proposed algorithm it is not possible to apply multiple test correction because the number of tests our algorithm performs makes it unrealistic to apply multiple test correction. Genome-wide association also

faces the same problem because of the large number of tests it performs [42]. The permutation test is a well known technique and has become the standard way of reducing false positive results in the haplotype pattern association study. We have applied this technique (see Section 4.3) in our algorithm to obtain a higher level of accuracy in the results.

Another cause of false positive results is the sample size of the cohort. There is a strong correlation between the statistical significance and the sample size. Small sample sizes may not represent the entire population and tend to produce false statistical significance results. This is a substantial risk when the result is interpreted as significant by chance. It is problematic to obtain a large cohort for such analysis; hence, the results of the smaller cohort should be interpreted carefully. Although there is no minimum sample size required for a cohort to be meaningful for statistical analysis, it has been proven that large sample sizes produce more accurate results [10, 41, 50].

In our algorithm we have used the haplotypes that are phased from the genotype data using the PHASE v2.0 algorithm. The accuracy of the produced haplotypes is an important factor for our analysis. The variability of the produced haplotypes from genotype data using different algorithms can affect any genetic analysis using haplotypes. The wrong haplotype or a large amount of missing data in the haplotype will severely hinder the accuracy of our algorithm. Different phasing algorithms produce different results with variable ranges in accuracy [33, 61]. We should be cautious in using phasing algorithms and notice the accuracy level each algorithm produces.

The proposed method of handling missing data should be revised for the use of

general haplotype analysis. This propose method integrates concepts of SNP linkage. The quality control filter that HapMap uses can be modified by integrating the proposed method of handling missing data because it will then allow the ignored data to be included in the HapMap study.

The proposed algorithm and its computation time implies that moderately large size SNP cohort data can be analyzed using this algorithm. However, we need to analyze more disease cohorts using this algorithm to verify the accuracy of the proposed CCGA algorithm. The population stratification bias is problematic to handle and it is hard to obtain an ethnically matched dataset. Hence, the control of stratification bias needs to be implemented in this algorithm. Future work on this problem should also be directed to perform a genome-wide association analysis using the proposed CCGA scheme. There is no existing method that can test genome-wide haplotype association. For a genome-wide association analysis, it may be feasible to use the proposed CCGA scheme where the genome can be decomposed into 23 species and each species will evolve an entire human chromosome. Given the increase in time complexity because the large number of SNPs will increase the search space exponentially, a parallel computing implementation of the CCGA scheme will probably be required. Additional speedups may be obtained by considering alternate (possibly heuristic approximations) of the various fitness-evaluation and collaboration mechanisms described in Chapter 4.

# Chapter 6

# Conclusions

We have proposed an algorithm that detects the susceptible SNPs and their underlying haplotypes for a complex disease using a populations case-control cohort. The algorithm uses a search method that allows detection of variable length haplotype patterns and the ability to detect such patterns from multiple genes simultaneously. The algorithm uses haplotype data that is obtained from various phasing algorithms and allows missing or unambiguous data in the haplotypes. The algorithm applies a variant genetic algorithm, namely, cooperative coevolutionary genetic algorithm (CCGA). The algorithm was applied to three different cohorts and the obtained results showed strong accordance with previously published results. The algorithm is specifically designed for an ethnically matched cohort and is not designed for genome wide-association.

The work presented in this thesis provides a technique for handling the missing or ambiguous data using the knowledge of LD structure of the chromosomal region. This technique may be applicable to any analysis using haplotype data. However,

the algorithm does not address the population stratification bias and that is one challenge for the future development of the proposed CCGA scheme. Handling this bias will allow the detection of SNPs and haplotypes for a complex disease in multiple ethnically distinct populations.

Current advances in genotyping technology allow us to genotype millions of SNPs of the entire human genome [58]. In a genome-wide case-control cohort it is still a challenge to perform haplotype association analysis due to the large number of SNPs. Our proposed CCGA scheme will need modification to scale up with such large amounts of data. One possible set of modifications is given in Section 5.5. Two additional types of modifications are – (*i*) parallelization of the CCGA and (*ii*) optimization of CCGA parameters and the statistical test parameters relative to whole-genome datasets. Techniques for parallelizing a CCGA can be adopted from previous literature [27]; however, how to efficiently optimize CCGA and statistical test parameters relative to very large datasets is an open problem.

# Bibliography


[1] R.M. Adkins. Comparison of the accuracy of methods of computational haplotype inference using a large empirical dataset. *BMC Genetics*, **5(22)**, 2004.

[2] M. Aoki-Suzuki. *et al.* A family-based association study and gene expression analyses of Netrin-G1 and -G2 genes in Schizophrenia. *Biological Psychiatry*, **57(4)**:382–393, 2005.

[3] K.G. Ardlie, L. Kruglyak, and M. Seielstad. Patterns of linkage disequilibrium in human genome. *Nature review Genetics*, **3(4)**: 299–309, 2002.

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

[5] P.I.W Bakker *et al.* Transferability of tag SNPs in genetic association studies in multiple populations. *Nature Review Genetics*, **38**:1298–1303, 2006.

[6] T. Blevins *et al.* Four plant dicers mediate viral small RNA biogenesis and DNA virus induced silencing. *Nucleic Acids Research*, **34(21)**: 6233–6246, 2006.

[7] D. Botstein, and N. Risch. Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease. *Nature Genetics*, **33**: 228–237, 2003.

[8] M.A. Brown,L.G. Kennedy,A.J. MacGregor, C. Darke, E. Duncan, J.L. Shatford, A. Taylor, A. Calin, P. Wordsworth. Susceptibility to ankylosing spondylitis in twins: the role of genes, HLA, and the environment. *Arthritis and Rheumatism*, **40(10)**:1823–8, 1997.

[9] S.R. Browning. Multilocus association mapping using variable-length markov chain. *American Journal of Human Genetics*, **78**: 903–913, 2006.

[10] L.R. Cardon, and J.I. Bell. Association study designs for complex diseases. *Nature Review Genetics*, **2**:91–99, 2001.

[11] L.R. Cardon, and L.J. Palmer. Population stratification and spurious allelic association. *The Lancet*, **361(9357)**:598–604, 2003.

[12] T.G. Clark, D.M. Lorio, R.C Griffiths, and M. Farrall (2006) Finding association in dense genetic maps: a genetic algorithm approach. *Human Heredity*, **60(2)**:97–108, 2005.

[13] P.J.P. Croucher *et al.* Haplotype structure and association to Crohns disease of CARD15 mutations in two ethnically divergent populations. *European Journal of Human Genetics*, **11**:6–16, 2003.

[14] M.J. Daly, J.D. Rioux, S.F. Schaffner, T.J. Hudson, and E.S. Lander. High resolution haplotype structure in human genome. *Nature Review Genetics*, **29**:229–232, 2001.

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

[16] C.T. Falk and P. Rubinstein. Haplotype relative risks: an easy reliable way to construct a control sample for risk calculations. *Annals of Human Genetics*, **51**:227–233, 1987.

[17] D. Fallin, A. Cohen, L. Essioux, I. Chumakov, M. Blumenfeld, D. Cohen, and N.J. Schork. Genetic analysis of case/control data using estimated haplotype frequencies: application to APOE locus variation and alzheimer's disease. *Genome Research*, **11(1)**:143, 2001.

[18] M. Fukasawa *et al.* Case-control association study of human netrin G1 gene in Japanese schizophrenia. *Journal of Medical and Dental Sciences*, **51(2)**:121–128, 2004.

[19] S.B Gabriel *et al.* The structure of haplotype blocks in human genome. *Science*, **296**: 2225–2229, 2002.

[20] J. Gelernter, H. Kranzler, J.F. Cubells. Serotonin transporter protein (SLC6A4) allele and haplotype frequencies and linkage disequilibria in African-and European-American and Japanese populations and in alcohol-dependent subjects. *Human Genetics*, **101(2)**: 243–246, 1997.

[21] D.E. Goldberg, and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G.J.E. Rawlings, editor. *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 69–93, 1991.

[22] D. Gusfield. An overview of combinatorial methods for haplotype inference. In *Proceedings of Computational Methods for SNPs and Haplotype Inference*, Lecture Notes in Computer Science, no. 2983, Springer-Verlag, 9–25, 2004.

[23] B.V. Halldorsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. Combinatorial Problems Arising in SNP and Haplotype Analysis. In *Proceedings Discrete Mathematics and Theoretical Computer Science*, Lecture Notes in Computer Science, no 2731, Springer-Verlag, Berlin, 26-47, 2003.

[24] E. Halperin, and E. Hazan. HAPLOFREQ - estimating haplotype frequencies efficiently. *Journal of Computational Biology*, **13(2)**:481–500, 2006.

[25] J.H Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

[26] J.P.A. Ioannidis, E.E. Ntzani, and T.A. Trikalinos. Racial differences in genetic effects for complex diseases. *Nature Genetics*, **36(12)**:1312–1318, 2004.

[27] T. Jansen and R.P. Wiegand. Sequential versus parallel cooperative coevolutionary $(1+1)$ EAs. In *IEEE Congress on Evolutionary Computation*, **1**:30–37, 2003.

[28] K.A. Jong and M.A. Potter. Evolving complex structures via co-operative coevolution. In *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, MIT Press, 307–317, 1995.

[29] K.A. Jong, and W.M. Spears. An analysis of the interacting roles of population size and crossover in genetic algorithms. In *Parallel Problem Solving from Nature*, Springer, **1**: 38–47, 1990.

[30] Y. Li, W-K. Sung, and J.J. Liu. Association mapping via regularized regression analysis of single nucleotide polymorphism haplotypes in variable sized sliding windows. *The American Journal of Human Genetics*, **80(4)**:705–715, 2007.

[31] J.D. Lohn, W.F. Kraus, and G.L. Haith. Comparing a coevolutionary algorithm for multiobjective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE Press, Los Alamitos, CA, 1157–1162, 2002.

[32] S.W. Mahfoud. *Niching methods for genetic algorithms.* Phd Dessertation, University of Illinois at Urbana-Champaign, 1995.

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

[34] A. Martinez-Mir *et al.* Genomewide scan for linkage reveals evidence of several susceptibility loci for Alopecia Areata. *The American Journal of Human Genetics*, **80(2)**: 316–328, 2007.

[35] C.E. Mathews *et al.* Genetic analysis of resistance to Type-1 Diabetes in ALR/Lt mice, a NOD-related strain with defenses against autoimmune-mediated diabetogenic stress. *Immunogenetics*, **55(7)**: 491–496, 2003.

[36] R.A. Mathias *et al.* A graphical assessment of p-values from sliding window haplotype tests of association to identify asthma susceptibility loci on chromosome 11q. *BMC Genetics*, **7(38)**, 2006.

[37] W.P. Maksymowych, P. Rahman, J.P. Reeve, D.D. Gladman, L. Peddle, and R.D. Inman. Association of the IL1 Gene Cluster With Susceptibility to Ankylosing Spondylitis - An analysis of three Canadian populations. *Arthritis and Rheumatism*, **54(3)**:974–985, 2006.

[38] M. Mitchell, J.H. Holland, and S. Forrest. When will a genetic algorithm outperform hill climbing? *Advances in Neural Information Processing Systems*, **6**:51–58, 1994.

[39] R. Nakamichi, S. Imoto, and S. Miyano. Case-control of binary disease trait considering interactions between SNPs and environmental effects using logistic regression. In *Proceedings of the fourth IEEE Symposium on Bioinformatics and Bioengineering*, IEEE Press, Los Alamitos, CA, 73–78, 2004.

[40] S. Nakamura, O. Ooue, K. Akiyama and K. Abe. Genetic polymorphism of complement C6 and haplotype analysis between C6 and C7 in a Japanese population. *Human Genetics*, **68(2)**:138-141, 1984.

[41] K.K. Nicodenus, W. Liu, G.A. Chase, Y.Y. Tsai, and M.D. Fallin. Comparison of type I error for multiple test corrections in large single-nucleotide polymorphism

studies using principal components versus haplotype blocking algorithms. *BMC Genetics*, **6(1)**:S78, 2005.

[42] D.R. Nyholt. Genetic case-control association studies-correcting for multiple testing. *Human Genetics*, **109(5)**:564–565, 2001.

[43] N. Patil *et al.* Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, **294(5547)**:1719–1723, 2001.

[44] M.A. Potter. *The design and analysis of a computational model of cooperative coevolution.* Phd Dessertation, George Mason University, 1997.

[45] M.A. Potter and K.A. Jong. A cooperative coevolutionary approach to function optimization,In *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, Springer-Verlag London, UK, 249–257, 1994.

[46] S. Purcell, M.J. Daly and P.C. Sham. WHAP: haplotype-based association analysise. *Bioinformatics*, **23(2)**:255-256, 2007.

[47] J.D. Rioux *et al.* Hierarchical linkage disequilibrium mapping of a susceptibility gene for Crohn's disease to the cytokine cluster on chromosome 5. *Nature Genetics*, **29**:223–228, 2001.

[48] G. Salanti, G. Amountza, E.E. Ntzani, and J.P.A. Ioannidis. Hardy–Weinberg equilibrium in genetic association studies: an empirical evaluation of reporting, deviations, and power. *European Journal of Human Genetics*, **13**: 840–848, 2005.

[49] A. Sawa, and S.H. Snyder. Schizophrenia: Diverse approaches to a complex disease. *Science*, **296(5568)**:692–695, 2002.

[50] D.J. Schaid. Power and sample size for testing associations of haplotypes with complex traits. *Annals of Human Genetics*, **70**:116–130, 2005.

[51] J. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology.* PWS Publishing Company, Pacific Grove, CA, 1997.

[52] J. Smith, and T.C. Fogarty. Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computation.*, IEEE Press, Los Alamitos, CA, 318–323, 1996.

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

[54] M. Stephens and P. Scheet. Accounting for Decay of Linkage Disequilibrium in Haplotype Inference and Missing-Data Imputation. *The American Journal of Human Genetics*,**76(3)**:449-462, 2005.

[55] M. Stoll *et al.* Genetic variation in DLG5 is associated with inflammatory bowel disease. *Nature Review Genetics*, **36(5)**:476–480, 2004.

[56] T. Strachan and A.P. Read. *Human Molecular Genetics 2.* John Wiley and Sons, New York, 1999.

[57] J.B. Summer. The isolation and crystallization of the enzyme urease. *Journal of Biological Chemistry*, **69(2)**:435–441, 1926.

[58] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature Review Genetics*, **449**:851-862, 2007.

[59] A.E. Timms *et al.* The Interleukin 1 gene cluster contains a major susceptibility locus for Ankylosing Spondylitis. *The American Journal of Human Genetics*, **75**:587–595, 2004.

[60] M. Uddin. *Efficient substring-pattern-based algorithms for analyzing SNP data.* Bachelor of Science (Honours) Dessertation, Memorial University of Newfoundland, 2005.

[61] M. Uddin, M. Sturge, C. Griffin, S. Benteau, and P. Rahman. Variability of haplotype phase and its effect on genetic analysis. In *Proceedings of 21st Canadian Conference on Electrical and Computer Engineering,Symposium on Biomedical Engineering (IEEE, CCECE)*, IEEE Press, Los Alamitos, CA, 595–599, 2008.

[62] M. Uddin, T. Yu, and T. Wareham. A cooperative coevolutionary algorithm for haplotype pattern detection in case-control data." In *Proceedings of the 3rd Annual Canadian Student Conference on Biomedical Computing*, 2008.

[63] M. Uddin, T. Wareham,P. Rahman, L. Peddle, W.P. Maksymowych, and T. Yu. A Robust Evolutionary Algorithm for Computing Significant Haplotype Patterns with Arbitrary Number and Distribution of Markers in Case-Control Data. In *Proceedings of the 10th International Meeting on Human Genome Variation (HGV2008)*, 23, 2008.

[64] J.C. Venter, D.M. Adams, G.G. Sutton, A.R. Kerlavage, H.O. Smith and M. Hunkapiller. Shotgun sequencing of the human genome. *Science*, **280(5369)**:1540–1542, 1998.

[65] J.C. Venter *et al.* The sequence of the human genome. *Science*, **291(5507)**:1304–1351, 2001.

[66] W.J. Welch. Construction of permutation tests. *Journal of the American Statistical Association*, **85(411)**:693–698, 1990.

[67] J Yang, and V. Honavar. Feature subset selection using a genetic algorithm. *Intelligent Systems and Their Applications*, **13(2)**:44–49, 1998.

[68] X. Yao. An empirical study of genetic operators in genetic algorithms. In *Microprocessing and Microprogramming*, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, 707–714, 1993.

[69] N. Yosef, Z. Yakhini, A. Tsalenko, V. Kristensen, A-L. Borresen-Dale, E. Ruppin, and R. Sharan. A supervised approach for identifying discriminating genotype patterns and its application to breast cancer data. *Bioinformatics*, **23(2)**:e91, 2007.

[70] G. Yong. Population size and sampling complexity in genetic algorithms. In *Proceedings of the Bird of a Feather Workshops (GECCO) :Learning, Adaptation, and Approximation in Evolutionary Computation*, Springer, 178–181, 2003.

[71] T. Yu, D. Wilkinson, J. Clark, and M. Sullivan. Evolving finite state transducers to interpret deepwater reservoir depositional environments. In *Proceedings of the IEEE World Congress on Computational Intelligence*, Hong Kong, 3490–3497, 2008.