Department of Computer Science
Series of Publications A
Report A-2010-3

Computational Methods for Detecting Large-Scale Chromosome Rearrangements in SNP Data

Jussi Kollin

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XII, University Main Building, on 25 October 2010 at twelve o'clock noon.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.helsinki.fi (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-6465-4 (paperback)
ISBN 978-952-10-6466-1 (PDF)
Computing Reviews (1998) Classification: G.3, I.6.8, J.3
Helsinki 2010
Helsinki University Print

Computational Methods for Detecting Large-Scale Chromosome Rearrangements in SNP Data

Jussi Kollin

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
jussi.kollin@cs.helsinki.fi
http://www.cs.helsinki.fi/jussi.kollin/

PhD Thesis, Series of Publications A, Report A-2010-3
Helsinki, October 2010, viii+197 pages
ISSN 1238-8645
ISBN 978-952-10-6465-4 (paperback)
ISBN 978-952-10-6466-1 (PDF)

Abstract

Large-scale chromosome rearrangements such as copy number variants (CNVs) and inversions encompass a considerable proportion of the genetic variation between human individuals. In a number of cases, they have been closely linked with various inheritable diseases.

Single-nucleotide polymorphisms (SNPs) are another large part of the genetic variance between individuals. They are also typically abundant and their measuring is straightforward and cheap.

This thesis presents computational means of using SNPs to detect the presence of inversions and deletions, a particular variety of CNVs. Technically, the inversion-detection algorithm detects the suppressed recombination rate between inverted and non-inverted haplotype populations whereas the deletion-detection algorithm uses the EM-algorithm to estimate the haplotype frequencies of a window with and without a deletion haplotype. As a contribution to population biology, a coalescent simulator for simulating inversion polymorphisms has been developed. Coalescent simulation is a backward-in-time method of modelling population ancestry. Technically, the simulator also models multiple crossovers by using the Counting model as the chiasma interference model.

Finally, this thesis includes an experimental section. The aforementioned

methods were tested on synthetic data to evaluate their power and specificity. They were also applied to the HapMap Phase II and Phase III data sets, yielding a number of candidates for previously unknown inversions, deletions and also correctly detecting known such rearrangements.

Computing Reviews (1998) Categories and Subject Descriptors:
G.3 Probability and Statistics: Probabilistic algorithms, Stochastic processes
I.6.8 Types of Simulation: Discrete event
J.3 Life and Medical Sciences: Biology and genetics

General Terms:
Algorithms, Experimentation

Additional Key Words and Phrases:
Inversion, Copy-Number Variation, Coalescent Simulation

Acknowledgements

This thesis would not have been written or finished if it was not for several people. My advisors, Prof. Heikki Mannila and Dr Mikko Koivisto, are the most important ones for providing the research topic and guidance, and significantly helping in the research that is culminated by the results found in this book. Together with Prof. Esko Ukkonen, they allowed me to keep working on this topic by arranging my funding by Helsinki Institute for Information Technology HIIT, the Algodan Centre of Excellence of the Academy of Finland, CompGenome project of the Academy of Finland and the Biosapiens EU project. The research in this book was done at the Department of Computer Science, HIIT and Algodan and in graduate schools ComBi, Hecse and FICS.

A lot of the quality that is in this thesis is due to my pre-examiners Dr Jaakko Peltonen and Dr Mikko Sillanpää. Other people who gave useful comments on the book were Jukka Kohonen, Esa Junttila, Dr Pauli Miettinen and Dr Arcadi Navarro. Further advice was given by several people, including, but not limited to, Dr Dario Gasbarra, Prof. Aapo Hyvärinen and especially Dr Stefan Schönauer. Nonetheless, all the errors left in this book are due to me alone.

I received immeasurable peer support from several past, present and future PhD students, Esa, Jaana, Janne, Jarkko, Jukka, Niina, Pasi, Pekka and Teppo to name but a few. I also need to thank the IT staff of our department and HIIT. In the summer 2010 alone, I spent over four decades of computing time on their computers.

Finally, I owe much to my parents and brother who have supported me without question throughout all my studies. Thank you, Risto, Kaarina and Kalle.

Contents

CHAPTER 1

Introduction

Bioinformatics is a field that has formed in the overlap of biology and information technology. The questions posed in bioinformatics are typically of biological nature, but the methodology blends both biology and computer science. The data-gathering rate in biology – in particular genetics, proteomics and other fields concerned with the functionality within the cell – has increased considerably due to the advances in laboratory techniques. For example, the first drafts of the human genome were released in 2001 [60, 134]. In 2007 another human genome sequence was published [74], and in 2008 eight human genomes and their differences were investigated [64]. Wheeler et al. [139], Bentley et al. [11] and Wang et al. [136] also investigated the human genomes of single individuals in 2008. The next step further is the *1000 Genomes Project*[1] with one of their goals being to sequence the genomes of at least 1000 people.

The larger the data sets are, the more beneficial automated methods for analysing them are. For instance, analysing the human genome without computers appears as, and most likely would be, a daunting task. Furthermore, the computations may be impossible in practice also with data sets of modest size, if the models applied to the data are particularly complex.

As troublesome as their handling may be, these large genome data sets enlighten us of a significant part of what contributes to defining what we are. Someday, the information accumulated this

[1] `http://www.1000genomes.org` (Accessed 02.11.2009)

way may help doctors to create personal medicine by taking into account the genetic factors of the patient. Overall, the desire to improve the quality of life is one loosely set goal for studying bioinformatics.

The genome of all humans is not identical but varies in several places. These parts having variations that are not shared by all are called *polymorphisms*. The different forms that appear are called *alleles*. If a polymorphism is present only in one form in a subpopulation or a data set, it is called *monoallelic* or *monomorphic*. *Single nucleotide polymorphisms* (SNPs), typically arising from point mutations, have been an important part of research for a number of reasons, in part because measuring them from people is relatively straightforward and cheap. As the name implies, in a SNP the nucleotide at one position may vary between people. There can be at most four different variants per position, as there are only four different bases in the DNA. Most of common SNPs, however, have only two variants. SNPs have especially been used in analysing the linkage between gene alleles that result in a notable change in the individual phenotype, i.e., observable characteristics of the person, such as a hereditary disease.

As the name implies, *large-scale rearrangements* involve larger segments of chromosome that are, e.g., translocated, deleted or inverted. In later chapters the focus will be on the latter two cases. A study that resequenced a diploid genome of an individual reported that with the inclusion of larger genetic variants than SNPs, the two copies of the same chromosome within an individual may have only 99.5% similarity [74] in terms of matching basepairs. Of the 12.3 Mb of variant basepairs they discovered in their study, 74% of them were due to non-SNP variation.

In part, the genetic difference between individuals is due to SNPs involving only single nucleotides, whereas *inversions* and *copy number variants* (CNVs) frequently involve several thousands of basepairs long segments, sometimes even millions of basepairs. CNVs are segments of the genome that appear in different numbers of copies in different people. These structural variants have been reviewed, e.g., by Sharp, Cheng and Eichler [109] and Feuk, Carson and Scherer [36]. Due to the extent of their contribution to genomic variance, it is therefore relevant to further investigate the effects, presence and formation of these rearrangements, as they

might play a larger than expected role in the diversity within the human species and between other species.

In many cases, these rearrangements have been found to be linked to a number of genetic diseases [110, 135]. Furthermore, they may help the speciation process, i.e., how a population genetically evolves into a new species [52, 66, 102]

Perhaps the most straightforward way of identifying the rearrangements is to resequence the genomes of a group of people, i.e., sequence chromosomes and compare the result with reference sequences to detect variants, although there are several lighter alternative methods, such as comparative genomic hybridization (CGH) (e.g., [26]) and paired-end mapping (e.g., [70]). Resequencing is an expensive process, especially if there is no guidance which areas of the genome to investigate. By comparison, genotyping SNPs is cheap and they are prevalent in most parts of the chromosomes. Hence, they are a readily usable tool for directing the attention of researchers to relevant areas of the genome by genome-wide association studies (e.g., [97, 114]) also when the underlying causes are not rearrangements.

This brings us to the core of this thesis. The thesis presents methods for detecting the potential presence of large-scale rearrangements from the human genome by means of whole-genome SNP data analysis. After presenting the methods, they are applied to real-world data sets, namely HapMap data sets to find putative regions of such rearrangements.

These questions have been addressed recently, for instance, by Bansal et al. [9] and Sindi and Raphael [112], who searched for inversions based on SNP data, and McCarroll et al. [81], Conrad et al. [20], Corona et al. [22] and Kohler and Cutler [68], who searched for deletions based on SNP data. This thesis builds the deletion-detection algorithms on the work of Corona et al., but some of the results, in particular the inversion-detection algorithm presented in this thesis, are the outcome of independent and parallel research from 2004 to 2009.

1.1 Some concepts of genetics and population genetics

The main focus in this thesis is on the human genome and, to a much lesser extent, on *Drosophila* genome. A *genome* represents the information inheritable from the progenitors of an organism. In the aforementioned organisms, these mean the *deoxyribonucleic acid* (DNA) located in the nucleus of the cell and in the mitochondria. Of particular interest are the *autosomes*, the chromosomes that are typically present in every human in two nonidentical copies in the nucleus. In humans there are 22 pairs of autosomes, as the sex chromosomes X and Y are not autosomes.

The chromosomes of eukaryote have *telomeres* and *centromeres*. In the classic drawing of a chromatid pair as an X, the part where the two chromatids are tied together is the centromere. The telomeres, on the other hand, are the ends of the chromatids. A chromatid, in turn, is an identical replicated chromosome in cell division that is tied together with its identical partner.

Strictly speaking, the term chromosome can be understood to mean not only the DNA sequence it contains but also the proteins bound to it. The scope of this thesis limits the model of a chromosome to a string of characters in a four-letter alphabet, each alphabet corresponding to one possible base. Because each base in DNA is typically bound to its counterpart in the same alphabet, these units are called *basepairs*.

While the study of the genome of one individual is interesting, so is the investigation of those of a population. *Population genetics* can be described as the field of studying the genetic composition of a population and how it changes over time. It focuses on questions such as how and why one trait gained frequency in a population. Such quetsions are tightly linked into natural selection and the theory of evolution. The approach taken is often a theoretical one, and some population-genetical models have become well-known in the field of bioinformatics. These models can be used in subsequent analyses of the history of the population. For example, estimating population histories based on genetic data often utilizes the coalescent in one form or the other as the population model. For instance, Alter et al. [3] use the coalescent to investigate the past population

size of gray whales where as Shapiro et al. [108] use it to reconstruct the population history of bisons in Beringia and come to a conclusion that bison population in Beringia likely had begun to shrink before the arrival of humans. The topic of the coalescent will be addressed in greater detail in Chapter 2.

SNP data are important for population genetics. They can be used to decipher aspects of the history of the population and the part of the genome the data are from. For instance, if the data are highly homogeneous, the population might have had a temporary decrease in size in the past, due to, e.g., an outbreak of a disease or famine.

SNPs are at the core of this thesis. The methods in this thesis consider only biallelic SNPs, i.e., SNPs that appear to have only two different forms, excluding the possible deletion allele (where the polymorphic nucleotide pair is not detected to be present). This covers a large part of all SNPs, because it is unlikely for the same basepair to undergo mutation twice.

As mentioned, humans typically have two copies of each autosome, both with one instance of each SNP in the chromosome. The alleles in different chromosomes usually cannot be measured separately. Thus, in the case of biallelic SNPs, the results can be written by using four values: two values for *homozygous* genotypes, i.e., both alleles are measured to be the same, one for the *heterozygous* genotype, i.e., the alleles differ, and one value for *no call*, or failed genotype call.

The data of the type described above is called genotype data. In genotype data it is not known which alleles are from the same chromosome. If we have inferred how the differring alleles are divided into the two parent-derived chromosomes and include this in the data, we call them *haplotypes* or haploid genotypes. Each haplotype is therefore the list of alleles in one chromosome. This process of inferring the assignation of alleles to different haplotypes is called *haplotype inference* or *phasing* and has been extensively researched [80]. In *trio data* – in which we have genotyped the triplet of the father, the mother and the child – this is easier than in data collected from *unrelated individuals*. There are three types of data relevant to this thesis: trio genotype data, genotype data from unrelated individuals and haplotype data. If the genotype data used to infer the haplotypes in the last case was composed of trios, the

1	1	1	0	0	1	1	1	0
1	1	1	0	0	1	1	1	0
1	0	1	0	0	1	1	1	0
0	0	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1	0
0	0	0	1	0	0	0	0	1

Figure 1.1: An example of a SNP data set with 6 haplotypes and 9 biallelic SNPs after encoding the alleles with ones and zeros.

haplotype data has only the parental haplotypes.

With this terminology, we can now define the format of the data we will use. For genotype data, the data set D is an $n \times m$ matrix; it has n individuals, represented by their genotypes, and m SNPs. Each element d_{ij} corresponds to the measured genotype of individual i in SNP j. For some notational simplicity in later chapters, we use also d_i^j to denote d_{ij}. If the data are haplotype data, we call n the number of haplotypes; thus, each individual is represented by two rows in the matrix. Because we consider only biallelic SNPs, the haplotype data set is a binary-valued matrix. Missing genotypes are typically in such case imputed based on the nearby SNPs.

By relabeling all SNPs to use only alleles '0' and '1' in the haplotype data, we lose some information on type of the SNP, but this information is not needed by the methods used in this thesis.

A small example of a haplotype matrix is given as Figure 1.1. In that, the fourth and sixth haplotypes have a different allele (0) in the first SNP than the other haplotypes.

In population genetics, as the name implies, modelling the population is necessary. The model can then be used to investigate how the population and its composition changes over time. A classic and well-used population model is the *Wright–Fisher model* [37, 143]. Essentially, the model assumes a constant-sized population of haplotypes, i.e., the haplotypes are not explicitly assigned into individuals. The mating is random, i.e., the parent of each haplotype in the preceding generation is sampled from a uniform distribution. Another assumption is that the generations are discrete, i.e., all of the previous generation dies out the moment all of the next generation is born. This haplotype population size is called the *effective population size*, and will be discussed further in Chapter 2. Note

that this is different from census population size, i.e., the number of haplotypes in the diploid population. In practice, there is no random mating. The effective population size is the size of a random-mating haplotype population that corresponds to the behaviour of the non-random-mating population. For a slightly more verbose introduction, see, e.g., Wiuf, Schierup and Hein's book [49, Ch. 1.4, pp. 11–17].

The *Hardy–Weinberg equilibrium* [47, 138] is a well-known principle about the distribution of genotype frequencies in a population. Let us assume that we have a SNP with only two alleles present in the population. Let these alleles be called 0 and 1 and their relative frequencies in the population p and $(1-p)$, respectively. Assuming the alleles are selectively neutral, i.e., they do not affect how many more offspring have the same allele in the next generation, and the mating is random, then over multiple generations the proportions of genotypes 00, 01 or 10, and 11 tend to p^2, $2p(1-p)$ and $(1-p)^2$, respectively. This state is called the Hardy–Weinberg equilibrium. Strongly deviating allele frequencies might suggest, for instance, that the alleles are not selectively neutral.

When talking about the frequency of a SNP allele in the population, the term *minor allele frequency* (MAF) is frequently used. Formally, MAF is defined as the relative frequency of the rarer SNP allele present in the sample in the case of a biallelic SNP. Thus, the range of MAF is $[0, 0.5]$ in a population.

If we investigate two SNPs at a time, the non-random dependency between the allele frequencies of these SNPs in a population is called *linkage disequilibrium* (LD) (reviewed, e.g., by Slatkin [113]). SNPs close to each other are not independent of each other but they are in *linkage*. Note that SNPs being in linkage disequilibrium does not always mean SNPs are in linkage. The level of LD due to linkage diminishes with increased distance between two SNPs. The computation of LD from a collected data set also requires that the haplotypes are known.

There are multiple different measures of LD for a pair of biallelic SNPs. Let us consider the cases where $p_{i,j}$ correspond to the relative haplotype frequencies of the first SNP being of allele $i \in \{0, 1\}$, the second allele being $j \in \{0, 1\}$ and the frequencies of the first and second SNPs being $p_{i,\cdot}$ and $p_{\cdot,j}$, respectively. Several different LD scores can be expressed by using these variables (see, e.g., [29]). In

this thesis, we limit ourselves to r^2, also known as Δ^2, a widely used measure of LD, which is defined as

$$r^2 = \frac{(p_{0,0} - p_{0,\cdot}p_{\cdot,0})^2}{p_{0,\cdot}\, p_{1,\cdot}\, p_{\cdot,0}\, p_{\cdot,1}}\,. \tag{1.1}$$

The range of r^2 is from 0 (independence) to 1 (complete correlation).

The decrease in LD is typically a result of *recombination* or crossing-over, in which material between the two non-identical chromosome copies is exchanged. In recombination during meiosis, the cell division process which produces haploid gametes (mature sperm and egg cells), the non-sister chromatids form a *chiasma*; these are the points where the non-sister chromatids exchange genetic information. This can be likened to cutting the chromatids at a point (the chiasma), exchanging the tails and then glueing them back together. If SNPs are close to each other, it is less likely for a recombination to occur between them and thus decrease their dependency.

Recombinations occur at a variable rate in the genome; this was reviewed e.g. by Kauppi et al. [63]. The *genetic distance* between two loci in the genome is measured with centimorgans (cM). One centimorgan represents the distance in which on average one crossover occurs once per 100 generations, or alternatively, the chance of one percent that a crossover occurs between the two loci in one generation. The *physical distance*, by comparison, is measured in basepairs, abbreviated as 'bp'. In this thesis physical distance is a more widely used concept than genetic distance. Larger denominations of physical distance are 'kb' and 'Mb' for a thousand and a million basepairs, respectively.

A large part of recombinations per generation occur in spots called *recombination hotspots* [5]. These hotspots are typically a few thousand basepairs long and separate regions of lower recombination rate. Regions of particularly low recombination rate often result in *haplotype blocks* in SNP data; we address these in greater detail in Chapter 3.

Not all SNPs are measured in a genotyping process. There are essentially two reasons for this: first, we might not be aware of the presence of the SNPs and second, we might choose not to genotype the SNP. Reasons for deciding to ignore a SNP include, e.g., high

similarity to other nearby SNPs, in which case the SNP would not add much information to the data set. This process of selecting SNPs representative for a region is called tag-SNP selection, for which there are several different algorithms (e.g., [17, 46]). The process that finds which nucleotides have SNPs in a population is called *ascertainment*; the process may not find all SNPs.

There are multiple different methods of discovering SNPs, one possible one being resequencing chromosome segments in multiple persons and listing the multiallelic loci as SNPs. We call this a *panel ascertainment scheme*, with the panel referring to the individuals whose genome was resequenced. As is apparent, the level of LD and the number of people for whom the resequencing is done can strongly affect the number and type of SNPs that are included in the sample. In effect, the SNPs with low MAF in the population are least likely to be found, but they are also the most common type of SNPs present in the genome according to a neutral mutation model (e.g., [124]).

1.2 Inversions

Inversion polymorphisms are large segments of a chromosome that occur reversed for a subpopulation [36, 53, 109]; known inversions in the human genome typically range in length from hundreds of basepairs to roughly 5 Mb. A basic illustration of an inversion is shown in Figure 1.2. The two strands in the figure represent different *arrangements*, i.e., different orders for the genetic material in the region.

The origins of different types of structural variation, including inversions, has been reviewed in [78]. One general mechanism that may result in inversions is called nonallelic homologous recombination or NAHR. In that, a recombination goes wrong because low copy repeats (LCR) are mistaken for each other and this results in wrong parts of chromosome being joined together in a recombination. Typically these result in segmental deletions or duplications. In the case of producing an inversion, the low copy repeats are inverted.

An alternate theory as for how inversions may come about is given by Ranz et al. [99] in *Drosophila melanogaster* and two re-

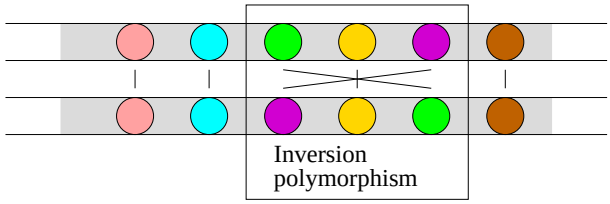


Figure 1.2: An illustration of two matched strands of non-sister chromatids with and without an inversion. The colours of the spheres correspond to homologous loci.

lated species, suggesting that the LCRs observed near inversion breakpoints are the result and not the cause of the changes resulting in an inversion.

A number of inversions have been associated with different human diseases [4]. It has also been argued that inversions and other chromosome rearrangements facilitate speciation, although the extent and method of their effect is uncertain [52, 66, 102].

A number of articles presenting previously unknown large-scale rearrangements in the human genome have been published, e.g., [1, 11, 64, 70, 74, 136, 139]. In some cases, the rearrangements were discovered by resequencing the complete genomes of at least one person. Complete resequencing of a genome is, however, still more expensive than whole-genome genotyping of an assay of SNPs. As mentioned, there are also other methods for detecting such rearrangements.

Originally, inversions were investigated in the genus *Drosophila*, in which a large amount of recent research pertaining to inversions has been done. For instance, according to Ranz, Casals and Ruiz [98], the rate of inversions fixed (so that the new arrangement becomes the only arrangement present) in the population per millions of years in the genus is estimated to be from 0.9 to 1.4 in the whole genome, which they report to be the highest rate found so far in eukaryotes.

Whereas haplotypes and genotypes involve single chromosomes, the characterization of all the chromosomes together is called a *karyotype*. In this thesis, the term appears only in reference to inversion homokaryotypes and heterokaryotypes, the former meaning

the individual has the inversion regions of both chromosomes of the same orientation whereas the latter means one chromosome has the inversion and the other does not. These are also frequently called homozygous and heterozygous for inversion.

From the perspective of this thesis, heterokaryotypes play an important role in the genetics of the human population. A significant part of the effects the inversions have in populations is how they affect recombinations. Let us consider an inversion heterokaryotype undergoing meiosis, i.e., a diploid cell dividing twice to produce four haploid cells. If there are no chiasmata formed within the inversion region, nothing happens differently from meiosis in homokaryotypes.

Regardless, let us now assume that one chiasma forms within the inversion region in a part of the meiosis process called prophase I. This situation is illustrated in Figure 1.3, where the different sequence orientations in a heterokaryotype result in the formation of an inversion loop. When the crossover is resolved and the chromatids move apart in anaphase I, the strand that was involved in a crossover has two centromeres which are pulled into different cells. The strand, a dicentric bridge, breaks, effectively leaving the two cells without a considerable portion of the arm of the chromosome. In practice, the cells that receive these remnants will not become viable gametes [94, pp. 242–244].

The inversions therefore effectively suppress recombinations within the inversion region in heterokaryotypes. Double recombinations, i.e., the ones with two chiasmata, within the inversion region can produce viable recombinants. Nonetheless, this is rather rare. Gene conversions are another method by which genetic material can be exchanged, although in such cases the gene conversion tracts, the genetic material that is exchanged, are typically much shorter [89] than what is exchanged in double recombinations.

This has certain effects on nucleotide variability within and near the inversions. These have been investigated in particular in *Drosophila* both by simulations or in theory [88, 89] and real-life experiments [87, 103]. The inversions have a greatly reduced gene flow, i.e., exchange of genetic material, between the two arrangements within the inversion; the effect is greater the closer the locus is to the nearest breakpoint.

Let us now define some further terms for use in later chapters.

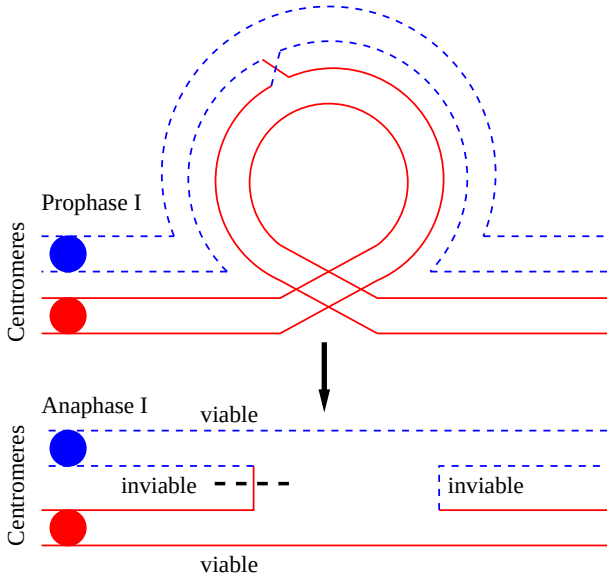


Figure 1.3: Part of meiosis with one chiasma in a heterokaryotype. The dicentric bridge resulting from the crossover is broken in anaphase I and the acentric fragment is lost. The picture has been adapted from [94, pp. 243]. Note that the upper and lower strands tied to the lower centromere have been exchanged in this figure in anaphase for readability.

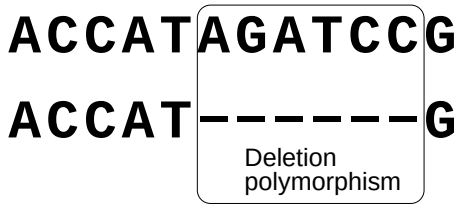


Figure 1.4: An illustration of chromosomes with and without a deletion.

Ancestral-type lineage/haplotype and *inversion-type lineage/haplotype* refer to two different arrangement orientations in an inversion region. These terms are used only in cases in which we know which is the ancestral orientation, i.e. the original prior to the inversion event. These are not typically used outside Chapter 2, as the ancestral orientation is often difficult to ascertain outside simulations.

To cater for the situations in which the aforementioned terms are not applicable, we use two other terms: *standard-type lineage/haplotype* and *alternate-type lineage/haplotype*. For inversions, the former use the orientation used in the human reference sequence or any other reference sequence by which the SNPs are ordered. The latter represents the lineages that have the orientation opposite to standard-type. In all cases, the types of such haplotypes are sometimes called *arrangements* due to the different order of homologous material inside them.

1.3 Deletions

In the case of a deletion polymorphism, a part of the chromosome is missing from some people, as shown in Figure 1.4. The length of this deleted part can vary greatly from one basepair to hundreds of kilobases and even larger. Deletions are a part of a larger group of polymorphisms known as copy number variants (CNVs). In the past couple of years, they have been extensively researched [135].

Deletions can affect the genes and their expression in a number of ways, but if the deletion is a short one, it may reside outside genes without affecting the genes and their expression levels. A

deletion within a gene can effectively prohibit the protein it encodes from functioning properly. If the deletion length in basepairs is not divisible by three, the rest of the codons in the gene are in practice nonsensical and do not bear a resemblance to the original gene. For example the gene allele resulting in blood type O is an allele of blood type A except for one missing nucleotide [144]. This is sufficient to effectively disable the gene.

Typically, deleted segments shorter than 1000 bp are considered indels, short for insertions and deletions [36, 135]. In this thesis, we do not make a difference between indels and CNVs, when the latter does not entail multiple copies of the locus.

Longer deletions may remove whole genes from the chromosome. For example, RhD negativity in Europe is often due to a complete deletion of a gene [8]. As with inversions, there are different diseases associated with deletions [131, pp. 274–280]. Many of them are not typically inherited diseases but due to *de novo* deletions, i.e., recurrent deletions that the parents themselves do not carry. The focus in this thesis, however, is on neutral deletions that are inherited from the parents, although the results for detecting deletions in unrelated individuals can also be used for identifying *de novo* deletions. This is conditional on the novel deletions affecting the same SNPs.

Furthermore, we focus on deletions that are typically longer than 1 kb. This is because we investigate only indirect evidence of the presence of deletions, more specifically SNPs, that typically are not closer than some hundreds of basepairs to each other even in dense data sets. The resolution of our method is therefore not sufficient for identifying shorter deletions. Hence, for instance, the one-basepair long deletion that resulted in blood type O would likely not be recognized.

1.4 HapMap data set

The International HapMap Project [127] has, in the past years, played an important role in bioinformatics. To quote the abstract of the publication describing the project [127], the goal of this project "is to determine the common patterns of DNA sequence variation in the human genome and to make this information freely available

in the public domain."

This goal is accomplished, e.g., by genotyping millions of SNPs over the human genome, phasing them to resolve the underlying haplotypes and estimating recombination rates across the genome. All these data are available on the project's website[2].

Because of the availability of the HapMap data, they have been used in several studies, (e.g., [9, 22, 68, 112, 126]). This thesis also uses the the HapMap data sets [128, 129] as real-world data for the methods presented in the previous chapters.

The data have been released so far in three phases: the first and the second phase contained SNP data from four subpopulations while the third phase added a number of other subpopulations. The third phase also increased the number of samples in some subpopulations.

The used HapMap data consist of the four populations that were present already in phases I and II: CEPH (people living in Utah with northern and western European ancestries, abbrev. CEU), people living in Yoruba in Ibadan, Nigeria (abbrev. YRI), Han Chinese from Beijing, China (abbrev. CHB) and Japanese in Tokyo, Japan (abbrev. JPT).

In phases I and II, the CEU and YRI data sets consisted solely of trios whereas JPT and CHB data sets contained only unrelated individuals. In phase III, JPT and CHB data sets still contained only unrelated individuals, but CEU and YRI data sets now included also duos (one parent and a child) and unrelated individuals.

1.5 Main contributions and organization

With the related biological concepts explained, we can now consider the main contributions of this thesis. The thesis focuses on presenting novel and improved methods for detecting large-scale rearrangements from SNP data and a detailed analysis of publicly available data sets.

- Chapter 2 introduces the theory behind simulating chromosome segments with paracentric inversions and a publicly avail-

[2] `http://www.hapmap.org` (Accessed 02.11.2009)

able coalescent simulator for that purpose. J.K. devised and implemented the simulator.

- In Chapter 3 a novel method of discovering potential inversion polymorphism regions from SNP data collections is developed. J.K. participated in developing the test score. This research was done in collaboration with Mikko Koivisto, Heikki Mannila and Leena Peltonen.

- In Chapter 4 we improve the time complexity of a haplotype frequency estimation method adapted by Corona et al.[22] for detecting deletions. J.K. participated in correcting the formulae of the efficient EM-algorithm and devised the method for determining the deletion end-points. This research is joint work with Jaana Wessman, Mikko Koivisto and Heikki Mannila. Preliminary work on the topic was done by Sanna Sipilä and Suvi Hiltunen.

- In Chapter 5 the methods described in Chapters 3 and 4 are first tested on synthetic data sets. HapMap Phase II and III data sets are then examined for inversion and deletion polymorphisms. The experimental setup was chosen mostly by J.K. with the exception of the deletion simulations, which was chosen as a subset of the experiments done by Kohler and Cutler [68]. All experiments were conducted by J.K.

Finally, in Chapter 6 we review the contributions of this thesis and discuss some future topics to pursue based on the results presented.

CHAPTER 2

Coalescent simulation of inversions

In this chapter, the basics of coalescent simulation are reviewed. A model for simulating multiple chiasmata in one recombination event in continuous time approximation is presented. It is shown how to incorporate a paracentric inversion model in coalescent simulation. Finally, the modelling of effective subpopulation sizes in case of an inversion is briefly considered.

2.1 Coalescent simulation

Due to its computational efficiency, the *coalescent* has become a widely used tool in theoretical population genetics ever since the introduction of Kingman's coalescent [65], a continuous-time approximation of the exact discrete time Wright–Fisher model. The coalescent process forms a tree similar to a phylogenetic tree of a segment of a chromosome, in effect being the genealogy for that segment. It is this trace of genetic material backwards in time to the single ancestor that is the coalescent. The coalescent has been used, for example, to estimate recombination rates [121] and to produce realistic synthetic SNP data sets [104] to measure the false positive rate of, e.g., deletion detection methods [68]. It has also been used with inversions [88] to estimate gene flow rates between different arrangements.

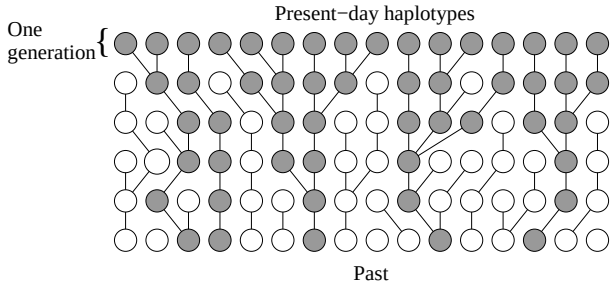


Figure 2.1: An example of a simple genealogy in a population of haplotypes. The gray nodes are haplotypes that are inherited by at least one present-day haplotype.

In this section, we briefly review the coalescent, the recombination model that extends the coalescent [55] and the backwards-in-time simulation of a chromosome segment.

The idea in the coalescent is to simulate n haplotype lineages backwards in time until the *most recent common ancestor* (MRCA) is found. The simulation is computationally efficient, and there have been several extensions to the basic model introducing functionality such as gene conversion [141], subdivided populations with migration (e.g., [49, 91]) and variable population size over time, reviewed for example by Donnelly and Tavaré [31].

See Figure 2.1 for an example of a small population of haplotypes without recombination. In it, the 16 present-day haplotypes of the population have only five ancestors five generations ago. Note, however, that this genealogy was not produced strictly according to the Wright–Fisher model assumptions, as the number of offspring for each haplotype in each generation was not sampled from the appropriate distribution.

The simulation can be carried out in two steps: first by generating the tree and then sampling the mutations in the tree branches. We now look at how the tree is constructed.

Let us first assume that all the generations are discrete, and the population in each generation is represented by $2N_e$ haplotypes, or N_e diploid individuals. We call N_e the *effective diploid population size*. In effect, accounting for diploid individuals is as simple as only multiplying the number of individuals by two. This is shown

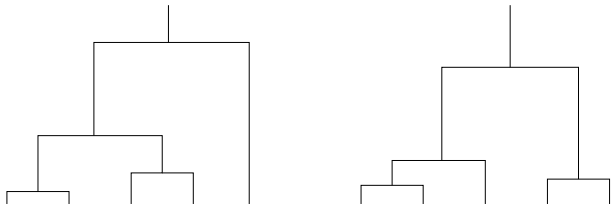


Figure 2.2: Two examples of coalescent trees of five samples. Each coalescence event is depicted as a joining of two branches.

in, for example, [91].

The Wright–Fisher model assumes random mating, i.e., the parent of each haplotype in the succeeding generation is sampled independently from all the haplotypes in the preceding generation. Ignoring the possiblity of recombination, each haplotype is a copy of one haplotype in the preceding generation. This means that any two haplotypes in the same generation have the same parent haplotype with probability $1/(2N_e)$. In such case, we say that these two lineages *coalesce* in the preceding generation and call this a *coalescence event*. These events define the genealogy of the simulated haplotypes. Note that the model is still defined in terms of random-mating haplotypes and not diploid individuals. In the terms of the latter, coalescing would mean that two children inherited the same haplotype from the same individual.

The generation of the genealogy can now be done by sampling the time for the next coalescence event, then randomly selecting two lineages and joining them into one, and repeating this until only one lineage remains: the MRCA. At this point the lineages have formed a *coalescent tree*, a binary tree where each non-leaf node represents a coalescence event. Figure 2.2 displays two coalescent trees for $n = 5$. Each leaf represents one haplotype in the sample and the root the MRCA. In essence, this is the phylogenetic tree of the haplotypes.

By simulating the genealogy backwards in time, we simulate only the necessary parts of the genealogy within the population. In forward simulation, some of the genetic material in the past generation may be lost before the present-day sample, which can mean unnecessary work in simulating the extinct lineages.

We now look at how the times for the coalescence events are sampled. From the Wright–Fisher model described here and in Section 1.1 the waiting time for the most recent coalescence event can be derived to be geometrically distributed with parameter $1/(2N_e)$, i.e., the probability of one pair of lineages coalescing t generations into the past is

$$\Pr(\text{coalescence at } t) = \left(1 - \frac{1}{2N_e}\right)^{t-1} \frac{1}{2N_e}.$$

To simplify the sampling of the waiting time until the next coalescence event and in the process eliminating the case of multiple coalescence events happening simultaneously to simplify the simulation, we instead approximate the discrete geometric distribution with the exponential distribution with the parameter $1/(2N_e)$ when N_e is large enough. The point density function for coalescence is now

$$f_{\exp}(t; \frac{1}{2N_e}) = \frac{1}{2N_e} e^{-\frac{t}{2N_e}}.$$

Furthermore, we rescale the time units from generations to $2N_e$ generations per unit and hence can use Exp(1) to model the waiting time for the first pair of lineages to coalesce. This has the effect of eliminating the effective population size from the sampling equations if the population size is constant.

In continuous-time simulations, each of the pairs of haplotype lineages coalesce independently. Let us denote by $\mathcal{L}(t)$ the set of lineages we are tracking at time t, i.e., the lineages that still need to coalesce before finding the MRCA. Now, when simulating all the $|\mathcal{L}(t)|$ lineages, the parameter for the exponential distribution is $\binom{|\mathcal{L}(t)|}{2}$ if the time scaling is $2N_e$ generations per one unit of time. The parameter comes from each pair of lineages coalescing independently. For now, we consider N_e a constant. As previously mentioned, it can also change with time, in which case we use $N_e(t)$ to denote the effective population size at time t. In such case, the time units are typically measured by $4N_e(0)$ or $2N_e(0)$ generations with time 0 corresponding to the present and time increasing into the past.

The simulation of recombinations is important for many applications of coalescent simulators that need to simulate parts of autosomes. In Hudson's model [55], recombination events occur at a

given rate for a lineage per generation. Each event models the formation of exactly one recombination breakpoint, and the recombining lineage always recombines with a lineage that is not currently in set $\mathcal{L}(t)$. We denote the recombination rate per generation in the whole simulated sequence by rl, where l is the simulated sequence length minus one (in base pairs) and r is the probability of a recombination occurring between any two adjacent bases per one generation. This notation assumes that the recombinations occur between any two adjacent bases at equal probability.

Similarly to the case of coalescence events, recombinations are assumed to happen independently of each other, which results in a geometric distribution for the waiting time for the first recombination event in generations. This waiting time distribution can also be approximated in continuous-time by the exponential distribution with parameter $rl|\mathcal{L}(t)|$, when the time is measured in generations. The intuitive meaning behind this parameter is the expected total number of recombination events in one generation.

Each recombination event in Hudson's model splits a lineage into two lineages that would have to be tracked in the simulation. This results in the tracked lineages having segments that are not inherited by any of the n haplotypes in the simulated sample. The rest of the genetic material in the two parent haplotypes formed another haplotype, but this was not any of the tracked haplotype lineages, which means it does not have offspring in the present-day sample. We call those segments that are inherited by the present-day sample the *ancestral material* of the lineage.

The introduction of recombination causes the haplotype histories no longer be described as trees, but as *ancestral recombination graphs* (ARG) [43], as lineages split by recombination may coalesce with other lineages before coalescing together again. Hence, the relationships between haplotypes are described as graphs rather than a single tree. The genealogy of any single position in the simulated segment can still be represented with a coalescent tree, which can be viewed as a subgraph of an ARG. Figure 2.3 shows a simple ARG where there has been one recombination at point 0.2 in the lineage of 'c'. The first part of the segment coalesced with the common ancestor of lineages 'a' and 'b' while the second part coalesced with lineage 'd'. The coalescent tree for the first part contains the arc labeled (1) but not the arc (2), whereas the coalescent tree for the

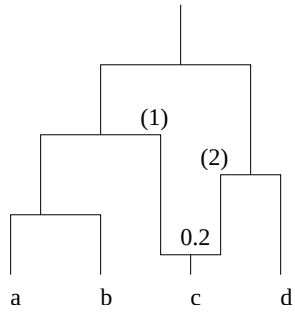


Figure 2.3: An example of an ARG with one recombination event in lineage 'c' at point 0.2.

latter part contains (2) but not (1).

To further simplify matters, we consider the *infinite-sites model*, in which the simulated segment is an interval of the real axis. Hence each crossover will always happen at a different position. A more important difference to the *finite-sites model* is that each mutation always occurs at a different place.

Let us now consider the division of the ancestral material by recombination more formally. Let the simulated segment correspond to $[0, 1) \subset \mathbb{R}$. The lineage in edge u of the ARG carries the ancestral material $a(u) \subset [0, 1)$. After a recombination has taken place in position $c \in (0, 1)$, the material is split into two lineages v and w so that $a(v) = a(u) \cap [0, c)$ and $a(w) = a(u) \cap [c, 1)$. An example of this is shown in Figure 2.4. Note that $a(v)$ or $a(w)$ can be empty; in such case, that lineage need not be simulated further.

The effects of a *gene conversion* greatly resemble a recombination event with two breakpoints. In a double crossover genetic material is exchanged but in a gene conversion the genetic material is copied from one DNA molecule to another. After a gene conversion event, the gametes are either like the original or they look like they had undergone a double crossover. Hence, gene conversions can be modelled as if they were double crossovers with different rates of occurrence and possibly different distributions for the breakpoints. However, the length of the gene conversion tract is typically considered different from what would be normal for two adjacent crossover points during the same meiosis. We use the model proposed by Wiuf

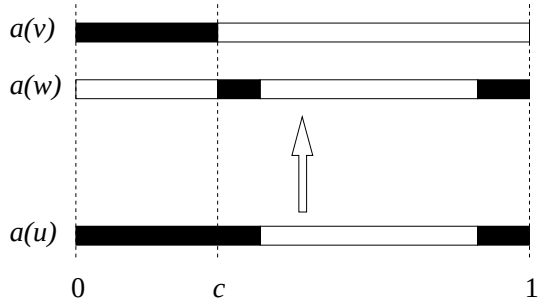


Figure 2.4: An example of recombination in Hudson's model. The simulated recombinational breakpoint c splits the ancestral material, marked in black, in lineage u into lineages v and w. White denotes material that is not inherited by any present-day haplotype. The simulation proceeds backwards in time, i.e., u and c determine v and w.

and Hein [141], in which the length of the tract is modelled by the exponential distribution.

After generating the complete ARG, the addition of neutral mutations into the model is straightforward. Each edge in the ARG has a length, measured in generations. Let us denote by μ the mutation probability of a single nucleotide per generation. By assuming mutations to happen independently at a certain rate per $2N_e$ generations, specifically $2N_e\mu(l+1)$ with μ being the mutation rate of one nucleotide per generation and $l+1$ the sequence length, we can sample the number and the positions of mutations that occurred in that edge; the former from a Poisson distribution and the latter from the uniform distribution, assuming constant mutation rate over the simulated segment. If we constructed the coalescent tree for the position of one simulated mutation, any sampled haplotype would have the mutated allele if and only if the edge that introduced the mutation was on the unique path from the leaf to the root. This is depicted in Figure 2.5 in the case of two mutations (the circle and the square in the coalescent tree). A white shape corresponds to the ancestral allele and a black shape to the new allele.

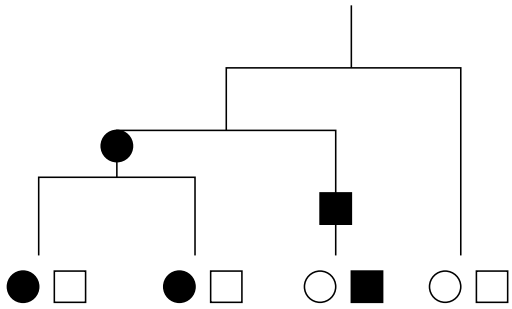


Figure 2.5: An example of how two mutations (the square and the sphere within the tree) simulated in the coalescent tree affect the present-day haplotypes. At leaves, the black shapes correspond to the new allele and the white shapes the ancestral allele.

2.2 Inversions in the coalescent

With certain adjustments, coalescent simulators can be used to simulate data containing inversion polymorphisms. The resulting framework resembles the one of Zöllner and von Haeseler [148] for disease gene simulation, with the greatest difference being in the interaction between the two subpopulations in recombinations. In both cases, the simulation consists of two separate subpopulations so that the lineages cannot coalesce across the subpopulation division. We track the set memberships of the tracked lineages; we call these sets lineage sets, one for ancestral-type lineages and one for inversion-type lineages, denoted by $\mathcal{L}_{\mathrm{A}}$ and $\mathcal{L}_{\mathrm{I}}$, respectively.

At some point of the simulation, i.e., after proceeding far enough backwards in time, one subpopulation has converged into a single tracked lineage in which the segregating mutation, in this case the inversion, occurs. This lineage, which corresponds to the original haplotype in which the segment was inverted, is then moved to the other subpopulation, after which the simulation continues as if there was only one subpopulation.

The paracentric inversion model, where the 'paracentric' denotes that centromere is not within the inversion region, is built on three rules:

1. The child haplotype inherits from its parent haplotype, or

haplotypes in the case of recombination, exactly one copy of each basepair. This excludes the meiotic products with two or no centromeres.

2. The inversion at this position is a unique event in the population history.

3. The chromosomes form chiasmata at the same rate in all individuals, regardless of their karyotype.

We also assume infinite-sites model: mutations and crossovers occur always at different positions and inversion end-points are never crossover points.

The first rule is implicit in coalescent simulation, but is stated here separately, as it is relevant when modelling recombinations with inversions. Sampling recombinations at their proper frequency is not as straightforward as it was without inversions due to reasons described in Section 1.2. The easiest way to do this is to suppress recombination events that would produce inviable meiotic products by rejection sampling. For a viable recombination as defined by rule 1, both inversion breakpoints in the recombination product must have been inherited from the same lineage set. This equals to either the recombination occurring in a homokaryotype or having an even number of simulated breakpoints within the inversion segment in a heterokaryotype. This is addressed in greater detail in Section 2.2.2.

In real-life genomes, e.g. in the human genome, inversions are in many cases recurrent events, but unique inversion events do exist as well [130]. We do not consider the former case here, as modelling the inversions as unique events suffices in many cases. Hence, rule 2 is not overly limiting while still simplifying the model. The rule could, however, be discarded by, e.g., modelling the repeated inversion events by the way of Zöllner and von Haeseler [148].

Let us denote by t_{I} the time at which the inversion event took place. Note that it is necessary that before the simulation passes t_{I}, the inversion population has found its MRCA, if the inversion-type haplotype population growth has been reasonably modelled.

If we needed to simulate only the segment enclosed within the inversion polymorphism, the inversion event can, for the most part, be represented as the birth of a new subpopulation at time t_{I}, and the inversion-type effective subpopulation size $2N_e^{\mathrm{I}}(t_{\mathrm{I}})$ being 1 at

the time of inversion formation, i.e., the subpopulation consists of only one haplotype. This simple simulation is not sufficient for all purposes, as it does not model gene conversion, double crossovers or increased LD due to suppressed recombinations. Also the effects to the LD levels near the breakpoints cannot be modelled this way.

Let us consider an implementation of the model and assume that parameters t_{I}, number of sampled haplotypes from both populations, and subpopulation growth models are given to the simulator. There are four base types of events in the simulation: an inversion event, coalescence events, recombination events and gene conversion events. Let us now review how these events are simulated.

The first event type is straightforward to simulate. At time t_{I}, all lineages in $\mathcal{L}_{\mathrm{I}}$ are coalesced into one lineage, which is then moved to $\mathcal{L}_{\mathrm{A}}$. Ideally the inversion lineage set should contain only one lineage at that time due to the inversion arrangement frequency approaching 1, as rule 2 assumed the inversion was unique and hence had a single progenitor.

The simulation of coalescence events remains straightforward. Because the inversion is modelled as a unique event, the coalescing is limited to only between lineages of same type. Even though the two haplotype populations coexist in the same diploid population, the time scaling on the coalescence events is the same as if the populations were completely independent. This is because the scaling for the waiting time is derived from the number of possible ancestors; each inversion-type lineage has as many possible parents in the preceding generation as is the inversion-type effective population size.

Hence, the lineages in the two sets coalesce independently of each other, and especially independently of the effective population size of the other set. The coalescing rate is therefore different, because the effective subpopulation sizes are most often not the same.

Recombination events are a more difficult case. To simulate them, the actual recombination event is replaced by a chiasma formation event, described next.

2.2.1 Multiple chiasmata formation model

The suppression effect the inversion polymorphism has on the recombination rates in the simulated segment is not straightforward.

As stated in various articles on recombination and gene conversion near and within inversion polymorphisms in *Drosophila*, double crossovers and gene conversions are important factors in modelling the gene flow [88, 89] between the two arrangements.

The frequently used Hudson's simulation model of recombinations [55] does not exactly simulate multiple crossovers within one generation, but instead creates multiple single recombination events within a narrow time frame. In the traditional setting, this difference is not as important as it is in simulating inversions, as the probabilities of double crossovers occurring are small, and these crossovers can still be decomposed into three separate events, two recombination and one coalescence, thus not completely prohibiting such events. The total probabilities of such events, however, are different from the actual probabilities.

Even ignoring this inaccuracy, the approach used in Hudson's model is insufficient for accurately simulating inversion regions: double crossovers cannot be accurately modelled for simulating inversions without considering them, or multiple crossovers in general, explicitly. This is because already the first step in the 'chained' model results in an inviable gamete that could not have been the parental haplotype.

To this end, Hudson's recombination model is here adapted for simulating multiple crossovers at a time, without giving up the continuous time approximation. We can see Hudson's model as the simulation of the formation of one chiasma. By comparison, the model presented here simulates the formation event of at least one chiasma, where the chiasmata are also not independently distributed within one generation. In this section, we consider only the case where there is only one population and no inversion present.

Similar to the parameterization of Hudson's model, let r be the probability of one recombination breakpoint forming in one generation between two adjacent bases assuming no inversion interference, as follows from rule 3. In the following, u denotes the lineage of the haplotype for which the crossovers are proposed and $\mathcal{L}_\alpha(t)$ the lineage set in which u is.

Note that another way of specifying r would have been to consider the recombination rate of the complete simulated sequence, and select the parameter conversion function from Hudson's model to the multiple chiasmata model so that the probability of the sep-

arate ends of the simulated segment recombining would be equal in the two models. In the following, we do not consider this approach of defining a model. Because double recombinations within segments elementary coalescent simulators can simulate are rare, the difference between the two parameters gained through different conversion functions would not likely be very large.

For simplicity, we do not consider chromatid interference models in which the probabilities of the participation of the four chromatids in chiasmata are not identical. We decide for each chiasma independently and with equal probabilities whether they affect the sampled chromatid or not. In practice this means that we pick one type of strand from the centromeric end of the simulated chromosome segment and track it to the telomeric end, switching the parental chromatid type with probability 1/2 whenever we come across a simulated chiasma.

This is not biologically accurate for all species; e.g. the results of Navarro et al. [89] show that in *Drosophila*, the chiasmata outside the inversion affect the proportion of produced viable gametes. However, this assumption simplifies equations.

We now take a look how the ancestral material is split in the case of multiple crossover breakpoints. Let us call the simulated breakpoints affecting the sampled chromatid $c_2 < \ldots < c_{m+1}$, and define the vector $c = [0, c_2, c_3, \ldots, c_{m+1}, 1]$.

Now, define

$$s_j^{\mathrm{A}}(c) \quad := \bigcup_{i=1}^{\lfloor (m+2-j)/2 \rfloor} [c_{2i-1+j}, c_{2i+j}), \qquad j = 0, 1. \tag{2.1}$$

The superscript A refers to the ancestral-type orientation. The case of inversion-type orientation is handled later. The function defines the subset that comes from parental chromatid of type j. Thus the two sets $s_0^{\mathrm{A}}(c)$ and $s_1^{\mathrm{A}}(c)$ are composed of interleaved disjoint intervals, with the crossover breakpoints being the interval end-points.

Note that $s_0^{\mathrm{A}}(c) \cap s_1^{\mathrm{A}}(c) = \emptyset$ and $s_0^{\mathrm{A}}(c) \cup s_1^{\mathrm{A}}(c) = [0, 1)$. With these alternating masks, the ancestral material in the parent haplotypes of lineage u are now $a(v) := s_0^{\mathrm{A}}(c) \cap a(u)$ and $a(w) := s_1^{\mathrm{A}}(c) \cap a(u)$.

One aspect of chiasma placement ignored by Hudson's model is chiasma interference or the model of dependence for the chiasmata

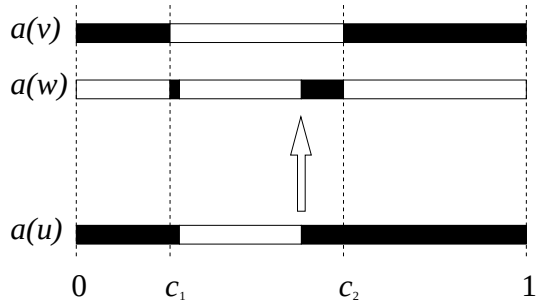


Figure 2.6: An example of multiple chiasmata in a simulated sequence.

formed in the same generation and chromosome. With at most one chiasma per generation, this is usually not modelled in coalescent simulation. For a larger view of genetic interference, see, e.g., [12].

There are several different chiasma interference models [15, 83]; we will discuss two of them here. In both cases the simulated tetrads, i.e., the structure of four chromatids formed in the prophase of meiosis, are assumed not to contain inversions.

Poisson model

The Poisson model [44] is arguably the simplest interference model, also called the no-interference model. Let us assume that the physical properties of the chromatids and chiasmata do not interfere with chiasma formation. Assuming infinite and independent possible crossover sites within the simulated sequence, the number of chiasmata in one generation is approximated by the distribution Poisson(λ); we show later in this section how we compute the parameter λ.

Each chiasma has the probability of 1/2 of affecting the sampled strand, because only chiasmata between non-sister chromatids leave a mark. Therefore, each of these chiasmata affects exactly one of the two sister chromatids of the type of the sampled strand. The probability follows from assuming both strands' involvement equally probable. Finally, the probability of having k crossover

breakpoints in the sampled strand with intensity parameter λ is

$$\Pr(k \text{ chiasmata}) = \sum_{i=k}^{\infty} \binom{i}{k} 2^{-i} f_P(i;\lambda)\,,$$

where $f_P(i;\lambda) = \lambda^i e^{-\lambda}/i!$ is the probability mass function for the Poisson distribution with value i and parameter λ.

It is now easy to state the expected number of breakpoints in the sampled strand per generation as

$$\begin{aligned}
\mathrm{E}[\#\,\text{chiasmata}] &= \sum_{k=0}^{\infty} k \sum_{j=k}^{\infty} f_P(j;\lambda) \binom{j}{k} 2^{-j} \\
&= \sum_{j=0}^{\infty} f_P(j;\lambda) \sum_{k=0}^{j} k \binom{j}{k} 2^{-j} \\
&= \sum_{j=0}^{\infty} f_P(j;\lambda) \frac{j}{2} \\
&= \frac{\lambda}{2}.
\end{aligned}$$

The second equality follows from standard manipulation of listing values of k and j in different order. The second to last step results as the expectation of a binomial random variable, and the last step is due to the expectation of a Poisson-distributed random variable.

With the expected number of chiasmata per generation specified, we can now solve λ from the equation so that the expected number of chiasmata per generation in the sampled strand is the same as it is for Hudson's recombination model, rl, i.e., $\lambda = 2rl$.

Because we are using exponential distribution to sample time to at least one chiasma to be present in the tetrad, we need to compute the probability for this event to serve as a parameter for the waiting time distribution. This parameter is

$$\begin{aligned}
\Pr(1 \leq \#\text{chiasmata}) &= 1 - \Pr(0 \text{ chiasmata}) \\
&= 1 - f_P(0;\lambda)\,,
\end{aligned}$$

which is simple to compute.

Finally, it is necessary to know how to sample the breakpoints from the conditional distribution under the condition of at least

one chiasma being present in the tetrad. Because the chiasmata are placed independently, it is possible to first sample the number of chiasmata from the tail of the cut Poisson distribution and then sample the positions of the chiasmata from the uniform distribution along the simulated segment. The chiasmata are placed along the simulated segment by a Poisson process. The sampling from the uniform distribution can hence be done as proven, e.g., in [86, Theorem 8.14].

Counting model

A well-known set of chiasma interference models is the Gamma family, in which the chiasma distances follow the Gamma distribution with shape parameter b and the scale parameter $1/\lambda$. The point density function of this distribution is

$$f_{\text{Gamma}}(x; b, 1/\lambda) = x^{b-1}\frac{e^{-x\lambda}}{(1/\lambda)^b\Gamma(b)}.$$

One particular subset of models, the Counting model [39], has the parameters as $b - 1 \in \mathbb{N}$ and λ. We focus on this model set for the ease of computation when generating chiasmata according to it. It has been a convention to label the former parameter of the Counting model as $m = b - 1$, but for the purpose of simplifying the equations in this chapter, we use b as the parameter instead.

An easy way of generating chiasmata according to this model in a fixed interval is to simulate successive points with distances from the exponential distribution with parameter λ. Starting from a randomly picked point of the first b points, we mark every bth point as a chiasma.

For *Drosophila*, McPeek and Speed [83] found that the best value for $m = b - 1$ was 3.94 under the Gamma model and not limiting only to integer values. Hence $b = 5$ can be seen as a good approximation of it. The same value has been reported to work well also for humans, as Lin and Speed [76] report.

We now derive the equations necessary to simulate the Counting chiasma interference model. In the model, we adjust the intermediate event distance distribution parameter λ so that the expectation of the number of chiasmata per generation matches that of the Hudson's model, rl. We use here the result that states that the

maximal number of terms in a beginning sequence of exponentially distributed random variables (distances between adjacent intermediate points) with total sum below a fixed threshold follows the Poisson distribution (the number of intermediate points) (e.g., [86, Theorem 8.7]). Thus we have

$$\begin{aligned}
\mathrm{E}[\#\,\text{chiasmata}] &= \sum_{i=0}^{\infty} i \Pr(i \text{ chiasmata in the sampled chromatid}) \\
&= \sum_{i=0}^{\infty} i \sum_{j'=i}^{\infty} \binom{j'}{i} \Pr(j' \text{ chiasmata in tetrad}) 2^{-j'} \\
&= \sum_{i=0}^{\infty} i \sum_{k=0}^{\infty} \sum_{j=0}^{b-1} f(j,k,\lambda) \left(\frac{b-j}{b} \binom{k}{i} 2^{-k} \right. \\
&\quad \left. + \frac{j}{b} \binom{k+1}{i} 2^{-k-1} \right) \\
&= \sum_{k=0}^{\infty} \sum_{j=0}^{b-1} f(j,k,\lambda) \left(\frac{b-j}{b} \sum_{i=0}^{k} \binom{k}{i} i 2^{-k} \right. \\
&\quad \left. + \frac{j}{b} \sum_{i=0}^{k+1} \binom{k+1}{i} i 2^{-k-1} \right) \\
&= \sum_{k=0}^{\infty} \sum_{j=0}^{m} f(j,k,\lambda) \left(\frac{b-j}{b} \frac{k}{2} + \frac{j}{b} \frac{k+1}{2} \right)
\end{aligned}$$

where $f(j,k,\lambda) = f_P(j+kb;\lambda)$ is the Poisson distribution probability mass function for parameter λ and value $j+kb$. The negative powers of 2 again follow from each chiasma having 1/2 chance of affecting the sampled strand.

To solve the corresponding scale parameter $1/\lambda$ of the Gamma distribution, this expectation is set to equal to rl. At this point, Newton's method can be used to find an approximate numerical solution for the equation. Because the probabilities of having high numbers of chiasmata in the simulated segment quickly become small, it is not necessary to compute the terms for high values of k to get an accurate estimate for λ.

We can then solve the corresponding waiting time for the first

chiasma that is in the tetrad from

$$\Pr(\geq 1\,\text{chiasmata}) \quad = \quad 1 - \sum_{j=0}^{b-1} \frac{b-j}{b} f_P(j;\lambda)$$

This probability has to be computed only once at the beginning of the simulation. We note that if $b = 1$, the above equations reduce to those of the Poisson model. Then the sampling of the actual breakpoints is done as for the Poisson model.

2.2.2 Chiasma formation and gene conversion events with inversions

The assumption of inversion events never co-occurring with chiasma formation events is made to simplify the assignment of the parent haplotypes in the lineage sets. For homokaryotypes, both ancestral lineages are set in the same lineage set, the same in which the child lineage was. In heterokaryotypes we set the lineage from which the child lineage inherited the inversion breakpoints to the same lineage set, and the other parental lineage to the other. This is because in the infinite-sites model the neighbourhood of the inversion breakpoints determine the orientation of the strand between them: no recombination can occur precisely at the inversion breakpoints.

However, there is one known problem relating to the use of the Counting model for which we do not present a solid solution. When placing the chiasmata along the interval $[0, 1)$, the chiasmata within the inversion loop in anaphase I need to be placed according to one orientation. Whichever orientation we choose, it is possible that the nearest chiasma outside the inversion is too close to the most distant chiasma within the inversion (Figure 2.7). Because multiple crossovers are rare within the sequences coalescent simulators can efficiently simulate, the chiasma assignment may be approximated by selecting the sequence orientation from the two parent haplotypes at random and use it for positioning the chiasmata in accordance to the Counting model.

The simulation of chiasma formation events in a population of both inversion-type and ancestral-type arrangements is done by a filtered Poisson process. In brief, because some potential meiotic products are inviable and thus result in an impossible child hap-

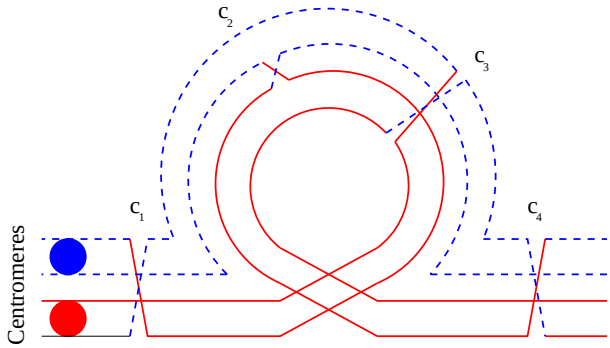


Figure 2.7: The simplest method of sampling chiasmata according to the Counting model is not accurate, as the chiasmata c_2 and c_3 both succeed c_1 as the next chiasma, making the sampling of distance between c_2 and c_3 problematic; likewise, both c_2 and c_3 precede c_4, presenting the problem on which preceding chiasma position to condition the position of c_4.

lotype, the filtered process will continue sampling until a feasible solution is found.

First, we model the probability of a lineage recombining in the absence of knowledge on the karyotype of the parent by dividing it in two cases: the parent is either a homokaryotype or a heterokaryotype.

In the case the parent is a homokaryotype, the recombination events can be directly simulated by the coalescent with the chiasma formation event intensity by sampling the positions of the chiasmata. In the case of heterokaryotypes, some chiasma formation events are to be rejected because of inviable resulting gametes. Based on rule 3 we set for the inversion simulation, we see that the observed total recombination rate in the whole population is larger than that within heterokaryotypes but lower than within homokaryotypes. This is because some of the recombinations that would be accepted in homokaryotypes are suppressed in heterokaryotypes.

We still need to consider the probability of the parent being a homokaryotype and the probability of the crossovers occurring in a heterokaryotype. Let us focus on the latter probability first. Consider a tetrad formed for a heterozygous genotype with respect to

an inversion. Because every valid haplotype has one centromere, we can pick any of the four centromeres as the point from where we start sampling chiasma positions. Because after the modifications due to recombinations and gene conversions only one of the chromatids in the tetrad is inherited by the offspring, we do not concern ourselves with crossovers occurring outside the one tracked chromatid.

We begin the tracking from the centromere. Each encountered crossover corresponds to changing the type of the source chromatid of the tracked haplotype, i.e., from alternative-type arrangement to standard-type arrangement or vice versa.

Let us assume that the strand we are following is originally of the standard arrangement when entering the inversion region. We maintain the assumption of not deleting or duplicating any material due to the crossovers, so each crossover makes the next chromatid segment to be read according to the other arrangement. The end of the inversion region is reached by following the strand in the standard arrangement if and only if an even number of crossovers were encountered within the inversion region. Otherwise, the strand is the inverted one, and the chromatid continues with the proximal region, which was already included in the chromatid. This breaks our rule of not including homologous genetic material multiple times. Also, by following the chromatid, we reach another centromere, forming a dicentric bridge. This bridge is then broken in anaphase I, resulting in two inviable gametes. In summary, we consider viable only those recombined strands that are either from a homokaryotype or have an even number of crossovers within the inversion region.

As mentioned earlier, this is not accurate for all species: e.g., Navarro et al. [89] state that the chiasmata outside the inversion affect the number of viable gametes in *Drosophila*. In practice, however, the discrepancy is small due to the unlikeliness of multiple crossovers.

We now address the issue of sampling the karyotype of the parent from which the recombinant haplotype was inherited. Recall that the karyotype here means whether the individual with the haplotype is heterozygous or homozygous with respect to the inversion. Let the recombination-produced lineage u be from the lineage set $\mathcal{L}_\alpha(t)$, where α stands for the orientation of the lineage. At least

one of the parent haplotypes for u must be of the same orientation as u. We name the haplotype of the same orientation as v and the other parent haplotype as w. Due to the random mating assumption, we can model the probability of the crossover being proposed in a homokaryotype by randomly selecting an unsimulated lineage from the joint population, i.e., a lineage that contains no genetic material inherited by the present-day sample. Let us name the type of the other parent haplotype as β. If we assume the joint effective population size to be the sum of the sizes of the two subpopulations, we can write the homokaryotype probability as

$$\Pr(\alpha = \beta \,|\, u \in \mathcal{L}_\alpha(t)) = \frac{N_e^\alpha(t)}{N_e^{\mathrm{A}}(t) + N_e^{\mathrm{I}}(t)},$$

where $N_e^*(t)$ is the effective population size of the corresponding subpopulation at time t.

If sampling from this gives that the parent is a homokaryotype, i.e., $\alpha = \beta$, or there is an even number of breakpoints within the inversion region, the chiasma formation event is accepted and processed. Otherwise, we sample the type of w and the number of recombination breakpoints again, this time permitting also the absence of chiasmata at time t. If the number of chiasmata is 0, the simulation continues with no changes. Otherwise, we return to checking the validity of the newly proposed crossovers and repeat, until a sampled strand is accepted or the sampled number of breakpoints is 0.

The reason why the number of chiasmata is allowed to be 0 only in the repeated samplings is because the sampled waiting time was for the event of at least one chiasma forming in the tetrad. This waiting corresponds to the exclusion of the case of 0 chiasmata in the first iteration.

For those chiasma formation events that are accepted, the resulting lineages that contain the inversion breakpoints will be added in $\mathcal{L}_\alpha$. The other lineage will be added in the same set as the hypothetical w was placed in.

Regardless of what the outcome of the rejection sampling is, the simulation progresses in time after the simulation time is increased by the simulated waiting time for the possibly rejected chiasma formation event. This is valid due to the waiting times being sampled from an exponential distribution.

As mentioned, this process defines a rejection sampling scheme, also called the acceptance-rejection method. Let us now investigate the situation closer and consider the space of all possible outcomes of transferring the genetic content from the parent to the offspring in the case of homokaryotypes. The probability distribution that the process described above defines over this space serves as the majoring function for the rejection sampling. In this case of rejection sampling, the point density function over regions where the suggested chiasma positions result in inviable gametes is set to 0. The resulting function is proportional to the desired distribution.

If recombination breakpoints are placed in an inversion-type lineage inside the simulated inversion, splitting ancestral material is slightly more complex, because the ancestral order in the MRCA differs from the physical order in inversion-type arrangements. An example of this can be seen in Figure 2.8(a).

We now look at the ancestral material division more closely. In the simulation, we keep track of only the inherited intervals but not the orientation of the intervals alone; the latter is handled by keeping track of the orientation of the inversion region. Let us denote the set of chiasmata affecting the sampled strand as c. The ancestral material masks $s_j^{\mathrm{I}}(c)$ for inversion-type arrangements can be computed in a way similar to standard-type arrangements, but the chiasma positions have to be transformed from standard-type to inversion-type arrangement. Let us denote the inversion region by $[b_s, b_e)$ and define a coordinate transformation γ as

$$\gamma(p) = \begin{cases} p, & p \notin [b_s, b_e) \\ b_s + b_e - p, & p \in [b_s, b_e) \end{cases} ; \tag{2.2}$$

in effect, we transform the 'physical' coordinates of the actual inverted sequence order into the standard order, which corresponds to the original order for the ancestral material. We do this for each point in the list of breakpoints c and then compute s^{A} for the transformed list (see Eq. (2.1)). Once we apply the inverse transformation to each point in the set of intervals s^{A}, we have $s_j^{\mathrm{I}}(c)$. An example of this is seen in Figure 2.8 (b).

If there is an odd number of recombination breakpoints within the inversion region, both s_0^{I} and s_1^{I} have one more contiguous subsegment. This results from the inversion breakpoints splitting one contiguous segment in two, which is relevant when simulating such

recombinations in inversion-type homokaryotypes.

It should be noted that in heterokaryotypes $a(u)$, the number of crossover points and their locations completely determine $a(v)$ and $a(w)$. In homokaryotypes, the division of ancestral matter in two lineages is also determined, but either of them can be called v or w.

Algorithm 2.1 summarizes the simulation of recombinations for coalescent simulation in the presence of inversions.

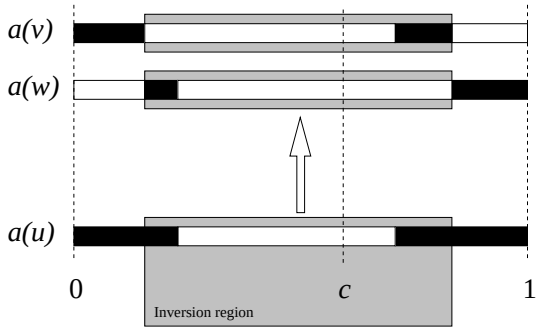
In our model, in heterokaryotypes gene conversions that involve a segment that contains either inversion breakpoint are prohibited to uphold rule 1 like in the case of double recombinations. Considering the shortness of the gene conversion tracts on average, this does not have very significant effects. We simulate gene conversion the same way as chiasma formation events with two or one chiasmata affecting the sampled strand, the latter number in the case either breakpoint is outside the simulated segment and hence does not affect the segment that is simulated. Note, however, that the distribution of the distance between the gene conversion breakpoints is different from that of two recombination breakpoints, unless the Poisson model with the same parameter λ is used.

Unlike in the case of crossovers, it is not necessary to simulate multiple simultaneous gene conversion events to maintain a reasonable level of accuracy, as the heterokaryotypes do not significantly affect gene conversion rates in our model.

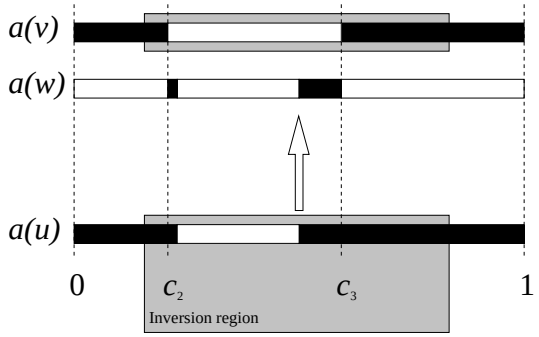
2.3 Modelling effective population sizes

It is important to specify the population model of the simulated population with reasonable accuracy to produce data sets that resemble real data sets. The population growth history affects, for instance, the minor allele frequency distribution of SNPs and the time until the most recent common ancestor, which will also be reflected in the number of simulated SNPs in the sample.

When simulating inversions, the population model is no less important. In addition to the aforementioned effects, if the inversion is a new one, it is likely to have less sequence variation within the inversion-type subpopulation than old inversions: assuming similar population growth models, the MRCA of inverted haplotypes is dis-



(a) Single recombinational breakpoint at c in an inversion homokaryotype.



(b) Double crossing over at c_2 and c_3 in a heterokaryotype.

Figure 2.8: Different types of crossover events with inversion-type lineages and how the ancestral material is divided between the parent haplotypes. Note that the inverted segments are represented in the ancestral order, not the physical, i.e., the regions on gray background are read from right to left and that the simulation progresses backwards in time. As a result, homologous content are lined up. Black represents the ancestral material lineage u inherits from lineages v and w.

Algorithm 2.1 Multiple chiasma formation event simulation with the inversion in the coalescent.

Input: The affected lineage type, $\alpha \in \{\mathrm{A}, \mathrm{I}\}$ and the associated lineage set $\mathcal{L}_\alpha$; inversion region end-points b_s and b_e

Output: Updated simulation status with the associated variables

Draw lineage $u \in \mathcal{L}_\alpha$.
$i \leftarrow$ *true* {To signal the first iteration of the loop}
while true **do**
  Sample $\beta \in \{\mathrm{A}, \mathrm{I}\}$, i.e., the type of w, the other chromosome in the parent, and c, the vector of chiasma positions in the tetrad
  if $i =$ *true* and $\#c = 0$ **then**
    Restart loop {$\#c$ refers to the number of elements in c}
  end if
  $c \leftarrow c$ thinned by removing each chiasma independently with probability $\frac{1}{2}$ {to omit chiasmata not affecting the strand}
  if $\#c = 0$ **then**
    Exit loop
  end if
  $i \leftarrow$ *false* {To mark the passing from the first iteration}
  Amend c with 0 and 1 at the appropriate ends of the vector.
  Select randomly either α or β, and if the selected type is the inverted type, then $c \leftarrow \gamma(c)$. {The chiasmata are placed according to an orientation present in the parent by applying the γ-transformation (Eq. (2.2) and (2.1)) to each element of c}
  $d \leftarrow \#\{c_j \mid c_j \in [b_s, b_e)\}$ {number of chiasmata within the inversion}
  if $\alpha = \beta$ or d is even **then**
    Create the ancestral lineages v and w
    if $b_s \in s_0^\alpha(c)$ **then**
      $a(v) \leftarrow s_0^\alpha(c) \cap a(u)$, $a(w) \leftarrow s_1^\beta(c) \cap a(u)$ {See Eq. (2.1)}
    else
      $a(v) \leftarrow s_1^\alpha(c) \cap a(u)$, $a(w) \leftarrow s_0^\beta(c) \cap a(u)$
    end if
    $\mathcal{L}_\alpha \leftarrow (\mathcal{L}_\alpha \setminus \{u\}) \cup \{v\}$, $\mathcal{L}_\beta \leftarrow \mathcal{L}_\beta \cup \{w\}$
    Exit loop
  end if
end while

covered sooner and there has not been enough time for mutations to be introduced in the inversion subpopulation.

The inclusion of varying effective population size affects in our simulation framework the coalescence event frequency. We now summarize the inclusion of changing population size as described in [49, Ch. 4.2].

As with non-time-homogeneous Poisson processes, the change in the effective population size can be taken into account by scaling the waiting time appropriately. Let

$$\Lambda(t) = \int_0^t \frac{N_e(0)}{N_e(u)} \mathrm{d}u,$$

the accumulated coalescent rate until time t with $N_e(u)$ denoting the effective population size at time u. Here, 0 represents the present and positive time units represent time before the present as in previous sections. With this, we can now express the probability of coalescing taking at least v time units, assuming the time now is t, with

$$\Pr(x > v \mid t) = \exp\left\{-\binom{|\mathcal{L}(t+v)|}{2}(\Lambda(t+v) - \Lambda(t))\right\}$$

where x is the random variable for the delay until the next coalescence event and $|\mathcal{L}(t+v)|$ is the number of simulated lineages at that time. Note that the larger $N_e(u)$ is compared to $N_e(0)$, the smaller the rate of coalescence is.

Let y be an exponentially distributed random variable with parameter $\binom{|\mathcal{L}(t)|}{2}$ and x the waiting time from the present simulation time, t. With this, we can solve x from the equation

$$\Lambda(t+x) - \Lambda(t) = y \tag{2.3}$$

to find the formula for sampling x with the help of y, i.e., the coalescing time of any pair of lineages in $\mathcal{L}$. The equation can be derived by using inverse transform sampling.

The frequently used exponential population growth model (e.g., [91], [49, Ch. 4.3]) is problematic in our case, because we need to model also the complementary population. Let us assume the inversion-type effective population follows the exponential growth pattern and that time is measured in units of $2N_e$ generations. In

this case, we can sample the waiting time to a coalescence event with

$$x_{\text{exp}} = \frac{1}{b_{\text{exp}}} \log\left(1 + b_{\text{exp}} y e^{-b_{\text{exp}} t}\right),$$

where b_{exp} is the growth rate parameter, y is again exponentially distributed with parameter $\binom{|\mathcal{L}(t)|}{2}$ and x_{exp} is the sampled waiting time.

The case of the complementary population is a difficult one. The simplest methods of solving the waiting time in such case are possibly numerical methods. More importantly, while the concept of exponential growth is reasonable in the case with one population, it might not be so with two subpopulations competing for the space within a joint population of constant size.

This directly ties to the question of how the joint effective population size is actually defined. One possibility is that the joint effective population size is the sum of the two effective subpopulation sizes and a constant. This principle has been used, for instance, by Navarro et al. [88]. Beside that, the method described earlier to estimate the probability of a heterokaryotypic person is based on the assumption that the joint population size is the sum of the subpopulation sizes but not necessarily a constant.

The connection between the subpopulations in equations is effectively reduced to determining the probability of a heterokaryotypic person, as the separation into subpopulations resulting from the inverted strand orientation carries over to also non-inverted regions. This results in considerable freedom in specifying how the subpopulation sizes change over time.

It should be noted that Zöllner and von Haeseler [148] used a fixed proportion for wild-type and mutant chromosomes in the population. Their simulation also did not require the original mutation to be unique. This is an alternative way to approach modelling two populations in the same location.

2.4 Implementation

A basic simulator for simulating inversions using the presented mod-

els has been implemented in Java and is available for download[1] and is called InvCoal. To simulate the generation of the ARG event by event, the simulator uses the idea used by Strobeck [120] and Navarro et al. [88] by sampling the waiting times for each event type, selecting the one with the smallest waiting time, updating the current simulation time and then resampling the waiting times for each event type anew with the possibly updated parameters.

The only subpopulation division the simulator supports is the division into inversion- and ancestral-type haplotypes. This limits the usability of the simulator. One subpopulation is limited to a constant-sized effective population whereas the other, the inversion type, has an exponential population growth model. Therefore the simulator does not use either the model of Navarro et al. [88] or Zöllner and von Haeseler [148]. The exponential growth model for the inversion population guarantees that the inversion population stemmed from a single haplotype.

The simulator uses the Counting model to simulate multiple crossovers. It is therefore important to evaluate the significance of this added functionality to the results compared to Hudson's coalescent simulator ms [57] under otherwise similar parameters, as Hudson's recombination model has been very widely used.

To test this, a constant diploid effective population size of 7,500 and recombination rates of 10^{-8} and 10^{-9}, denoting the average number of recombinations per basepair per generation, were used. Because the double recombinations become more frequent with higher recombination rate, the effect of varying it is relevant for evaluating the estimation difference between the two models. The mutation rate was constant at 10^{-8} per bp per generation. The used chiasma interference parameter for the Counting model in InvCoal was $m = 4$ ($b = 5$).

Because both simulators model recombination rate as a constant over the simulated segment, it is justifiable to compute the average r^2 (see Eq. (1.1)) over SNPs at a specific distance apart. To strengthen the data signal, SNPs with minor allele frequency under 0.05 in the total sample of 500 haplotypes were removed. The simulated segment was 1 Mb in length. Figure 2.9 depicts the behaviour

[1]The program is available at `http://www.cs.helsinki.fi/u/jkollin/software/InvCoal` (Accessed 02.11.2009)

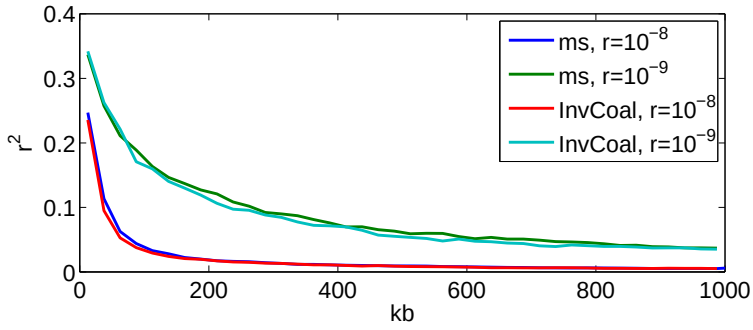


Figure 2.9: Average LD measured by r^2 over a specific distance of SNPs for two coalescent simulators; bin size is 25 kb. The parameter r in the legend is the recombination probability per generation and basepair. No inversion was simulated.

of mean r^2 in 100 repeated samples. The simulated segment was split into 25-kb-long bins and the first bin was chosen as the reference. Each bin was paired with the reference bin and the mean r^2 between all cross-bin SNP pairs was plotted.

In the figure we see that the LD levels between ms and InvCoal are slightly different but mostly similar. When the interference parameter was set to 100, effectively eliminating the possibility of having multiple chiasmata, the difference between the curves was smaller (data not shown). This is because with the segment length of 1 Mb, double recombinations with the used recombination rates would be very unlikely, effectively making the recombination model into a single-chiasma model.

To experiment on the need of our simulator for simulating inversions, we used the same simulation parameters to compare the simple inversion model, i.e., no gene flow between the arrangements and the subpopulations not interfering with recombination at all, with the model described in Section 2.2.1. To investigate the effect of recombination suppression on the levels of LD, two sets of experiments with different inversion ages were run. In both cases, the modern-day inversion type haplotype population size was $2 \times 0.3 \times 7500$. Figure 2.10 displays the behaviour of r^2 in such case; there were 500 haplotypes of both inversion- and ancestral-types. The increased LD in the inversion population in InvCoal is due to the modelled

recombination suppression within the inversion. Note that the simple model does not suppress recombinations the way InvCoal does, because in the simple model all individuals are effectively assumed to be homokaryotypes.

Finally, we briefly investigated the signal previously used to detect inversions [9, 112]. The signal was essentially having higher LD levels than expected at a distance away in a subpopulation. The signal is visible in Figure 2.11. The reference bin was set at 275–300 kb, which placed it just outside the inversion, which spanned the region of 300–650 kb. The visibility of this pattern of LD suggests that this simulator could be used to evaluate the performance of the said inversion detection algorithms under controlled scenarios. Because ms cannot reproduce this signal, InvCoal was not compared to it in this experiment setup.

Of some interest is the higher LD in the inversion subpopulation outside the inversion. This is likely due to the smaller effective population size in the inversion population. The lineages in the inversion population coalesce faster in comparison to the ancestral-type population, and hence there are fewer branches to recombine.

The reason why more complex simulators were not used in comparisons was to better identify the effects of the differences in the used models. The ms program can easily be parametrized to be similar to InvCoal, so the differences in the outputs are likely to follow from differences in the model and not the other additional features the simulator entails.

2.5 Discussion

We have presented a framework for multiple crossovers in coalescent simulation. The model is for the most part similar to the model described by Navarro et al. [88, 89] but ignores the details in gametogenesis, i.e. the process of generating gametes, and selection. There are also differences in the modelling of tetrads, as the model described in [89] was adjusted for Drosophila. The model presented there was extended to permit more than two chiasmata within the simulated region, although having such things occur within segments that a coalescent simulator can simulate is unlikely. In Section 2.2.1 a way to convert recombination rates to

simulation with the Counting model was shown.

Our model also results in that heterokaryotypic individuals have fewer offspring than homokaryotypic ones due to the rejection sampling scheme. It can be criticized that such large rearrangements are liable to be show the hitchhiking effect of the inversion gaining in frequency due to the favouring of nearby beneficial gene alleles. This effect has been used in coalescent simulators by, e.g., Braverman et al. [14]; Navarro et al. [88] adapted this model to their use in simulating inversions.

As chiasma interference models we considered the no-interference model (Poisson model) and the Counting model, a subclass of the Gamma family of models and a generalization of the Poisson model. The Counting model was selected as the chiasma interference model mostly because of the reasonable trade-off between accuracy and simplicity of implementation. There are several other chiasma interference models, but in experiments, the Gamma model has been found a good option [15, 83]. However, Housworth and Stahl [54] report that the detected double recombinations in a human data set are best described with a mixture of Gamma and exponential distance distributions, the latter corresponding to the no-interference model. The interference parameter also seems to vary between chromosomes and sexes of the same organism. The current version of InvCoal does not include this model.

There are several other coalescent simulators available. Of interest in the future chapters are COSI [104], the parameters of which have been calibrated to produce SNP data as seen in human autosomes. The simulator models varying recombination rate across the simulated segment and recombination hot spots. However, it has the same problems when simulating inversions as ms has, as it cannot suppress recombination in a subpopulation only on a short segment (the inversion) like InvCoal can. There is also another very recent simulator for simulating population genetic data with inversions [92]. InvCoal was developed independent of such very recent other simulators.

InvCoal is expected to be a helpful tool in investigating the effect that the inversions have on LD just outside the inversion regions. In potential future studies, the simulator may also be able to answer to questions pertaining to the changes in SNP MAF and LD distributions under different inversion population growth models.

However, the MAF distribution alone is likely to be insufficient for detecting the presence of inversions. Other potential uses are present where coalescent simulators in general are used. As an example, haplotype inference software typically assumes that adjacent SNPs are in higher LD due to linkage than remote ones. An inversion can turn this situation the other way, and the effect this has on haplotyping accuracy has not yet been investigated.

The Counting model as described here cannot completely accurately model recombinations in heterokaryotypes because of the chiasma interference from two different directions near the inversion breakpoints. The mathematically sound solution for this would be to model the distances between SNPs with conditional Gamma distributions. The current setup was chosen for its simplicity.

Many modern coalescent simulators model varying recombination rates and recombination hot spots. The Counting model is readily useable with these extensions.

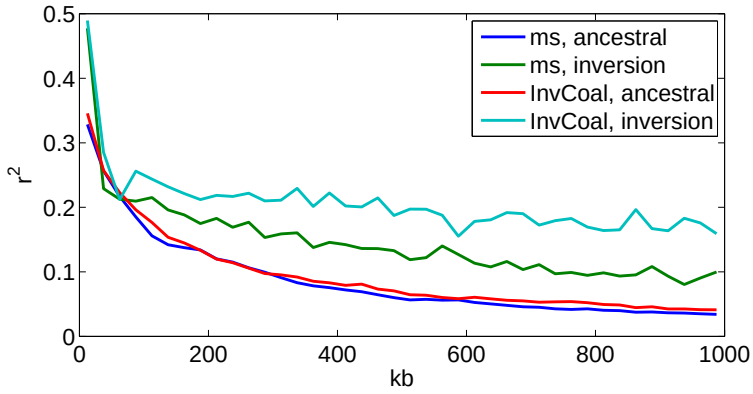
The used model of gene conversion can also be criticized. Schaeffer and Anderson [103] report in their experimental study that heterokaryotypes appeared to have reduced rate of gene conversion events near the inversion breakpoints. This aspect is not simulated by our adaptation of the gene conversion model of Wiuf and Hein [141].

Nevertheless, it is important to address the case of rejection sampling for viable gametes. In the model presented we resample the parent karyotype if the suggested recombination is rejected. This results in simulated semisterility of heterokaryotypes. If we had only sampled the parent karyotype once and then repeatedly sampled the chiasmata for the simulated region, the situation would correspond to random mating where the partners are fixed until the offspring is produced and the number of children is sampled from the same distribution for both homo- and heterokaryotypes due to, e.g., cultural reasons. Such a reason could be, for instance, monogamy, although this is in contradiction with the random mating assumption of the Wright–Fisher model.

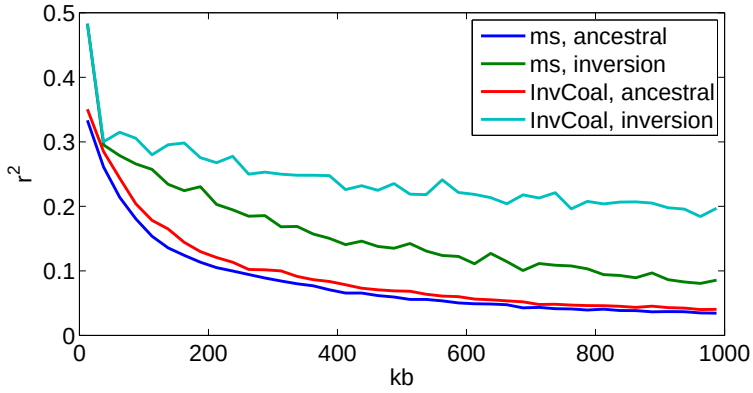
On the other hand, not all inversions are underdominant, i.e., they result in lesser infertility amongst heterokaryotypes [23]. In such case, not resampling the karyotype would be a crude approximation of simulating non-underdominant inversions.

One can ask if the inversion model could be used to estimate

the age of an inversion based on the SNP data akin to coalescent genealogy samplers, reviewed by Kuhner [72]. For every data set generated by the simulator with a fixed population model we can compute its likelihood. Fully investigating this option would likely require a careful revision of the inversion and population models presented in this chapter.



(a) Inversion age 40,000 generations.



(b) Inversion age 150,000 generations.

Figure 2.10: Average LD measured by r^2 over a specific distance of SNPs for two coalescent simulators and different simulated subpopulations; bin size is 25 kb. Inversion population size in present was $2 \times 0.3 \times 7500$.

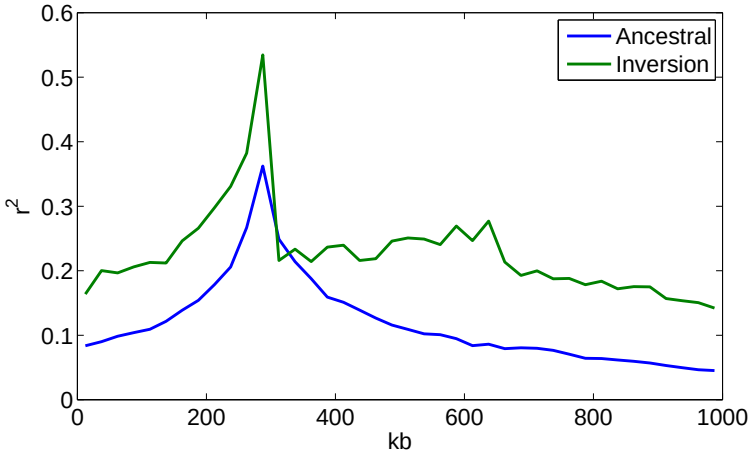


Figure 2.11: Average LD measured in InvCoal simulations by r^2 over a specific distance of SNPs; bin size is 25 kb. The simulations had an inversion at 300–650 kb.

CHAPTER 3

Detecting inversions

Inversions leave a footprint in the SNP data by suppressing recombinations within the inversion polymorphism region. We examine this footprint and how it can be used in detecting inversions and their breakpoints.

3.1 Inversion signals in SNP data

When SNPs are genotyped, their positions along the genome are not measured relative to the genotyped strand. The positions of the SNPs are instead reported in reference to a reference sequence, such as the NCBI RefSeq collection [96]. Hence, if the DNA elements are permuted in a different order in some individuals, the SNPs are typically not listed in the physical order of these individuals.

In the reference sequence, the SNPs within an inversion polymorphism region are ordered according to one of the orientations. Recall that the order in the reference sequence is called the *standard-type arrangement* and the other the *alternate-type arrangement*. The difference to the ancestral-type and inversion-type arrangement used in Chapter 2 is that we are now unaware which is the ancestral orientation of the segment and which one is the novel orientation.

In practice, an inversion polymorphism in SNP data of m SNPs can be defined as a region spanning SNPs a through b, if the physical order of those SNPs in one physical haplotype is $(a, a+1, \ldots, b-1, b)$ and for at least one other haplotype $(b, b-1, \ldots, a+1, a)$. Note

that the order of the SNPs in the SNP data set is the same for both types. In fact, an individual may even have both orientations present. Throughout the chapter, however, we assume that we know the haplotypes underlying the genotypes.

This means also that inversions are not necessarily evident in SNP data sets. However, inversions leave evidence, here called a signal, in the population genetical data by promoting unusual LD patterns and suppressing recombinations. This signal can be observed in at least two ways: the signal left by the breakpoints, and the signal within the inversion.

We first consider the former signal. Let us consider the case of four SNPs $s_1, \ldots, s_4$ so that s_1 is barely outside the inversion region in the proximal end of the inversion, s_2 is barely inside the inversion region near s_1 and the SNPs s_4 and s_3 are set analogously in the distal end, respectively. We also assume that the recombination rate is constant across the fictitious inversion.

Now, in standard arrangement the SNPs s_1 and s_2 are in higher LD than s_1 and s_3 because s_1 is closer to s_2 than s_3. The same applies to the pair s_3 and s_4 compared to the pair s_2 and s_4. In the alternate arrangement, however, this is not so. The physical distance between s_1 and s_3 is now shorter than that between s_1 and s_2, because the physical order of s_2 and s_3 has been inverted.

This signal has been used by Bansal et al. [9] and Sindi and Raphael [112] to discover putative inversion breakpoints. The method of the former is limited to the case where the standard-type arrangement is the rarer arrangement of the two, but the latter overcome this. A visualization of this LD signal is illustrated in Figure 2.11.

Whereas the signal at the polymorphism region boundaries resulted from the difference in the physical and genetic distances, the signal within the inversion regions originates from the inversions suppressing recombinations in heterokaryotypes [18, 122]. In heterokaryotypes, recombinations with one chiasma within the inversion region result in inviable meiotic products (e.g. [94, pp.242–244]). The gene flow between the two arrangements is not completely suppressed, though, as double crossing overs and gene conversions can still shift genetic material across the division.

We now focus on modelling this characteristic by two assumptions resulting in a simplified model. As in Chapter 2, the first simplifying assumption is that the inversion event is unique and

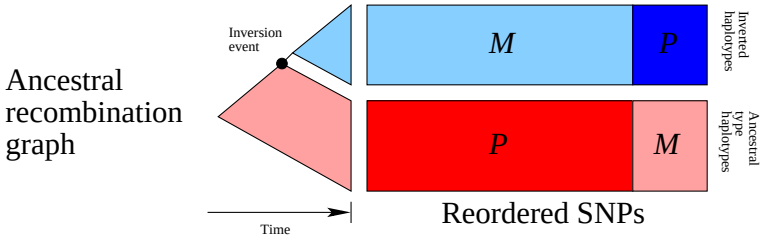
happened exactly once in the population history. Second, we suppress all recombinations and gene conversions in heterokaryotypes within the inversion region. This likely is not very a unrealistic assumption, because in the model described in Chapter 2 double recombinations in a 1-Mb-long segment are rare. The exclusion of gene conversion, however, is a more serious limitation, but gene conversions typically apply for shorter segments than double recombinations. This makes the impact of their exclusion smaller.

From these assumptions it follows that SNPs in the inversion region that are polymorphic in the population of one ordering are always monomorphic in the other population. Which population the SNP is polymorphic in can differ for each SNP, as explained next. For SNPs that are introduced by mutations after the inversion event took place, the novel allele cannot move from one arrangement to another under the assumption of no cross-arrangement gene flow, leaving the other population with only the ancestral allele. For SNPs that were introduced before the inversion, with the assumption of a unique inversion event, the inversion arrangement population consists of exactly one haplotype with exactly one allele of each SNP. Figure 3.1(a) displays the genealogy of the inversion region in this model.

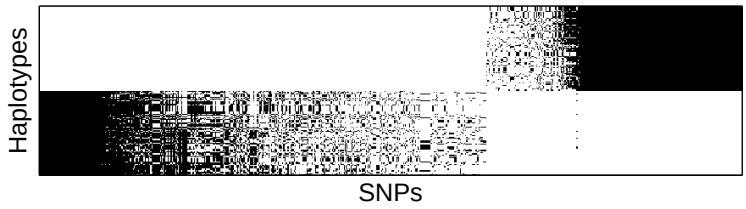
Let us assume that in this scenario, we can first group the haplotypes so that inversion-type haplotypes are separated from those of ancestral-type, and then sort the SNPs in the inversion region by their minor allele frequencies in the two haplotype populations. Ideally, this would result in a pattern similar to what is shown in Figure 3.1(b). We call this the *four-field pattern*.

While this model is an oversimplification, it is still likely that recent inversion polymorphisms have not had sufficiently many double recombinations or gene conversion events to significantly lessen this effect. For instance, the 900-kb inversion polymorphism, which is common only in Europeans [117], fits the pattern well, as shown in Figure 3.2.

In this light, a measure of how well the SNPs in a chromosome segment can be fitted into the four-field pattern can still be used as a signal for putative inversion polymorphisms. Unfortunately, haplotype blocks [25, 41, 146] can also be fitted well into the four-field pattern, and we have to distinguish inversion footprints from haplotype blocks. We consider this task next.



(a) Schematic of the ancestral recombination graph and the resulting data set inside the inversion in the simplified scenario. P corresponds to polymorphic SNPs and M to monomorphic SNPs.



(b) SNP data, which was generated by InvCoal, after sorting rows and columns conveniently. Regions of a single colour correspond to monomorphic SNPs.

Figure 3.1: The different signals inversions leave in synthetic SNP data. The data was produced by the simulator described in chapter 2.

Haplotype blocks are regions where the haplotypes can effectively be divided in few distinct, internally highly homogeneous groups. There are several different definitions for these blocks: regions where the average D', which is another LD measure, is above a threshold given as a parameter [101] and a low number of distinct haplotypes covering a majority of all haplotypes in a long region [25, 93]. In these regions, the recombination rate is typically below the average, as frequent recombinations would break the block structure, i.e., haplotypes being near-identical within the block of multiple haplotypes and multiple SNPs within a limited region. It has been observed that recombination hotspots seem to coincide with haplotype block boundaries [62].

A difference between haplotype blocks and inversion polymorphisms is that while recombinations are rare in the latter case, they are not completely suppressed. Especially in the ancestral-type set

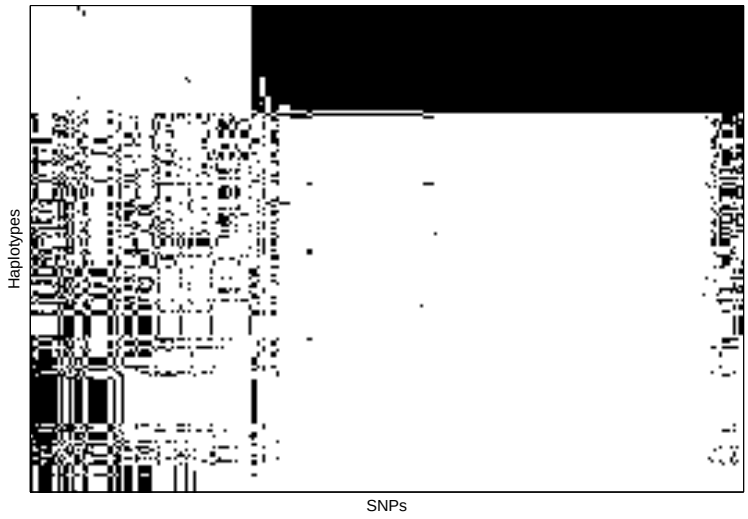


Figure 3.2: The SNPs in the HapMap Phase III [129] CEPH (CEU) data set (rel. 2) in chr17:40,899,921–41,989,253, which covers a known inversion. Each row corresponds to one haplotype, and each column corresponds to one SNP. The SNPs and haplotypes both are sorted to display the four-field pattern.

of haplotypes recombinations are typically visible because of the accumulated recombinations prior to the occurrence of the inversion event. Hence, we can sometimes detect traces of recombinations as pairs of SNPs that pass the four-gamete test, i.e., all four allele combinations of two biallelic SNPs are present in the data.

3.2 Normalized bicomponent score

We now develop a score for detecting inversions from SNP data by the reduced gene flow. Let us consider the simplified model from Section 3.1, i.e., there are no double recombinations or gene conversions. We approach the detection of putative inversions by assuming we have found a good division of the haplotypes in two putative groups: standard- and alternate-type and need to evaluate the goodness of the division. The details of how to infer the partition are considered later in Section 3.3.

The completely suppressed recombinations and gene conversions in heterokaryotypes mean that all SNPs that are polymorphic within the inversion polymorphism are monomorphic in exactly one of the haplotype populations. This pattern is visible in long windows of multiple SNPs. In short windows it is more likely for the window to display such signal by chance. If the window were short in basepairs, LD would be expected to be strong and thus haplotype variability reduced. If the window were short in the number of SNPs, finding a good bipartition of the haplotypes to display the signal would be more likely.

To measure this signal we devise a scoring with specific requirements. It needs to be quick to compute to accommodate to whole-genome analysis with dense SNP data sets. With this in mind, a composite marginal likelihood approach (reviewed in [133]) called independence likelihood that assumes the SNPs to be independent is easy to compute and hence suitable.

Let us have SNP haplotype data D with n haplotypes and m SNPs, and d_i^s denote sth measured SNP in the standard order in ith haplotype. We assume all SNPs to be biallelic, the two alleles being 0 and 1, i.e., D is a binary-valued matrix.

The inversion-detection scheme consists of solving two subproblems. First, it is necessary to find the putative division between the two arrangements. Second, this bipartition has to be scored based on how well the assumption of no cross-arrangement gene flow fits the division.

Let us investigate the latter question first, and assume we are given a bipartition of the haplotypes of D and name these two sets I and N. The former corresponds to the alternate-type arrangement haplotypes and the latter to standard-type arrangements.

We devise the scoring by modelling the data as a mixture of two components or probability distributions that are joined together as a convex combination. One component models the SNPs biallelic in I but monomorphic in N, while the other has the situation reversed. In mixture modelling, the distribution of one data point is a convex combination of the component distributions. In this case, the mixture model becomes

$$\begin{aligned}\Pr(d^s|\mathrm{N},\mathrm{I}) &= q\cdot\Pr(d^s_{\mathrm{N}}|s \text{ biallelic in I})\Pr(d^s_{\mathrm{I}}|s \text{ biallelic in I})\\ &+ (1-q)\cdot\Pr(d^s_{\mathrm{N}}|s \text{ biall. in N})\Pr(d^s_{\mathrm{I}}|s \text{ biall. in N})\,.\end{aligned}$$

Here q and $1-q$ represent the mixture proportions, d^s_{N} and d^s_{I} refer to the data at the sth SNP within groups N and I and d^s is the vector containing all the observed values of the SNP s, i.e., the joined d^s_{I} and d^s_{N}.

If SNP s is biallelic in the other group I, we assume that it is nearly monomorphic in N and vice versa. The word 'nearly' is used because genotyping errors and errors in phasing can break the strict monomorphicity, not to mention the double crossovers and gene conversions.

First, the SNP alleles are modelled as a sequence of Bernoulli-distributed random variables. Because the actual parameter for the distribution, i.e., the relative frequency of one allele, is unknown, the Bayesian approach is used by marginalising over the parameter space. For an introduction to Bayesian data analysis, see, e.g. [42].

Let us define a Beta prior distribution, $\mathrm{Beta}(\alpha, \beta)$, for the frequency of allele 1 and denote this frequency as θ. By assigning the hyperparameters α and β the same value between 0 and 1, values of θ near 0 and 1 are favoured, which corresponds to favouring nearly monoallelic SNPs.

With this prior distribution, the total probability of the observed data over all values of θ can be computed. The density function for Beta distribution $\mathrm{Beta}(\alpha, \beta)$ is

$$f_{\mathrm{Beta}}(\theta; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1},$$

where the normalizing constant, ensuring that the total probability equals unity, is

$$\frac{1}{B(\alpha, \beta)} = \frac{1}{\int_0^1 z^{\alpha-1}(1-z)^{\beta-1}\mathrm{d}z} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)},$$

and $\Gamma(y)$ is the Gamma function, a generalization of the factorial $(y-1)!$ to real numbers.

The likelihood of the data given frequency parameter θ is then given by that of a sequence of Bernoulli-distributed random variables. Let us denote by a and b the number of observations d^s_i, for $i \in \mathrm{N}$, that equal to 0 and 1, respectively.

By integrating θ out analytically, we get with standard formula

manipulation

$$\begin{aligned}
&\Pr(d^s_{\mathrm{N}}|s \text{ biallelic in I}) = \int_0^1 \Pr(\theta)\Pr(d^s_{\mathrm{N}}|\theta)\mathrm{d}\theta \\
= &\int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \times \theta^a(1-\theta)^b\mathrm{d}\theta \\
= &\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times c \times \int_0^1 \frac{1}{c}\theta^{\alpha+a-1}(1-\theta)^{\beta+b-1}\mathrm{d}\theta \\
= &\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times \frac{\Gamma(a+\alpha)\Gamma(b+\beta)}{\Gamma(a+\alpha+b+\beta)} \\
&\times \int_0^1 \frac{\Gamma(a+\alpha+b+\beta)}{\Gamma(a+\alpha)\Gamma(b+\beta)}\theta^{\alpha+a-1}(1-\theta)^{\beta+b-1}\mathrm{d}\theta \\
= &\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times \frac{\Gamma(a+\alpha)\Gamma(b+\beta)}{\Gamma(a+\alpha+b+\beta)} \times 1\,.
\end{aligned}$$

The variable c corresponds to $B(a+\alpha, b+\beta)$, which is introduced so that we can state that the integral sums to unity as an integral over a Beta distribution. Analogously, we define the corresponding probabilities for monoallelic SNPs within the inversion group I, that is, $\Pr(d^s_{\mathrm{I}}|s$ biallelic in N).

The probability of data, assuming polymorphicity in the population, is simpler to model. The appropriate prior parameters for θ are now $\alpha = 1$ and $\beta = 1$ since we do not want to favour any particular proportions *a priori*. These values result in a uniform prior distribution. With such assignments, we note that the factor $\theta^{\alpha-1}(1-\theta)^{\beta-1}$ in the previous equation becomes unity and can be removed from the equation. Hence, we have

$$\Pr(d^s_{\mathrm{N}}|s \text{ biallelic in N}) = \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)}\,.$$

It is well-known that SNPs near each other are not independent. However, we make this assumption to considerably simplify the model and computations, and combine the probabilities of the SNPs in the window into a composite marginal likelihood as the product

$$\Pr(D|\text{components N and I}) = \prod_{i=1}^{m} \Pr(d^i|\mathrm{N},\mathrm{I})\,, \tag{3.1}$$

where D refers to the entire haplotype data in the window and m is the number of SNPs within it. The assumption of independence

also bypasses the problem of the physical order of SNPs varying in different arrangements.

Eq. (3.1) measures how well the bipartition fits the model, but by itself, it is insufficient to measuring the benefit of using a bicomponent model for the data. Therefore, this two-component model is compared with a one-component model, for which the probability of the joint data set is simply

$$\Pr(d^s|\text{one component}) = \prod_{s=1}^{m} \frac{\Gamma(a_s+1)\Gamma(b_s+1)}{\Gamma(a_s+b_s+2)};$$

here a_s and b_s are the numbers of 1s and 0s in the data for the sth SNP. Here, we have set the hyperparameters for the Beta prior as 1 and 1, in which case the prior distribution places no preference on any particular allele frequency.

Note that no assumptions that would make the labels N and I unexchangeable were used in the analysis. This means that we do not need to specify which haplotype set represents the standard arrangement to use this model, which makes the task of finding the bipartition {N, I} easier. In less words, cluster identification is not a problem.

Definition 3.1 *(Bicomponent Score, BS) Given a bipartition {N,I} of haplotype data D,* bicomponent score *is defined as*

$$\text{BS}(D|N,\ I) = \log\left[\frac{\Pr(D|\textit{components}\ \text{N}\ \textit{and}\ \text{I})}{\Pr(D|\textit{one component})}\right],$$

i.e., the Bayes factor on the logarithmic scale.

Let us consider this from an information-theoretical viewpoint. If we used base-2 logarithms, we notice that the scoring can be written as

$$\log_2 \Pr(D|\text{one component})^{-1} - \log_2 \Pr(D|\text{components N and I})^{-1}.$$

Here, the first term is the number of the bits required to encode the data with one component, and the second is the same for encoding the data with two components [107]. Thus, the base-2 BS has the interpretation of the number of bits saved by encoding the data with two components instead of one.

This interpretation reveals one flaw in the bicomponent score: it is sensitive to the complexity of the window, as the encoding length of one window is likely to vary.

An intuitive solution to this is to normalize BS with the one-component data description length, eliminating the effect of the description length of the data.

Definition 3.2 *(Normalized Bicomponent Score, NBS) For a window of SNPs, we define the* normalized bicomponent score *as*

$$\mathrm{NBS}(D|\mathrm{N},\mathrm{I}) = \frac{\mathrm{BS}(x|\mathrm{N},\mathrm{I})}{-\log \Pr(D|\textit{one component})} .$$

The interpretation for NBS is now the proportional saving in the data description length we gain by using two components instead of one.

We still need to specify the values of the mixture proportion, q, the hyperparameters α and β, and the bipartition $\{\mathrm{N},\mathrm{I}\}$. If we set $\alpha = \beta = 1$, we get a particularly simple variant in which the mixture proportion is cancelled and does not have to be specified.

The score presented is still robust to genotyping errors and mislocated SNPs: the former because of the Bayesian model, which marginalizes the model over all values of θ, and the latter because all the SNPs are considered independent. The score also favours two haplotype groups over one as desired, when such groups are present.

3.3 Finding the subdivision between arrangements

To use the scoring presented in Section 3.2, the bipartition $\{\mathrm{N},\mathrm{I}\}$ of the data set x needs to be specified. Informally speaking, a good bipartition is one in which each SNP is biallelic in exactly one of the groups.

While the details of the method for finding this division is not important for NBS, for the method to be useable in whole-genome analysis, it has to be efficient and return at the very least a good approximation of the best division.

By using synthetic data, several different methods for splitting the haplotype set in two were experimented on:

- by sorting the haplotypes according to spectral ordering (e.g., [7, 38, 115], described also shortly below) and then splitting the sequence in two parts by testing all possible division points and selecting the division point that maximizes NBS with $\alpha = \beta = 1$,
- as above, but with $\alpha = \beta = 0.1$,
- k-Means [79] with a Manhattan distance metric with $\alpha = \beta = 1$,
- k-Means++ [6] with a Euclidean distance metric with $\alpha = \beta = 1$,
- k-Means++ with data points sorted according to the proportion of their Euclidean distance from one centroid, i.e., $v_i = \mathrm{d}_2(d_i, c_1)/(\mathrm{d}_2(d_i, c_1) + \mathrm{d}_2(x_i, c_2))$, where c_1 and c_2 are the two centroids, $\mathrm{d}_2(x, y)$ the Euclidean distance between x and y, d_i is the ith haplotype and v_i the representative value of d_i by which the haplotypes are ordered. Then all n splits are tried as with spectral ordering. In this case, this approach is called 'k-Means++ ordering'.

In all cases that used k-Means and k-Means++, the number of clusters was set to 2. The feature vectors used by these two methods to cluster haplotypes were the binary vectors telling the haplotype alleles. In the cases where the informative prior was used, i.e., $\alpha = \beta = 0.1$, which gives the prior distribution a non-constant form, the best value for the mixing proportion q was selected from the set of $0.1, 0.2, \ldots, 0.9$ as the one to maximize NBS for each investigated subpopulation division separately.

Let us now briefly cover the essentials of spectral ordering in this application. Informally speaking, the haplotypes are sorted so that similar haplotypes are set next to each other, after which we find the division by assuming the first haplotypes in the sorted set form N, and the last haplotypes form I.

More formally, given two haplotypes d_i and d_j, with indices $i, j \in \{1, \ldots, n\}$, we define their similarity $S(i, j)$ as the number of SNPs

s for which the alleles d_i^s and d_j^s agree. The $n \times n$ Laplacian matrix L of S is defined by

$$\begin{aligned} L(i,j) &= -S(i,j) \quad \text{for } i \neq j\,, \quad \text{and} \\ L(i,i) &= -S(i,i) + \sum_{j=1}^{n} S(i,j)\,. \end{aligned}$$

The matrix L is said to be positive semidefinite in, e.g., [115], so its eigenvalues are real and nonnegative. The smallest eigenvalue is 0, and the eigenvector $(z_1, \ldots, z_n)$ corresponding to the second smallest eigenvalue minimizes the function

$$\sum_{i=1}^{n} \sum_{j=1}^{n} S(i,j)\,(z_i - z_j)^2 \tag{3.2}$$

under the constraints $\sum_i z_i = 0$ and $\sum_i z_i^2 = 1$ (see [7]).

Thus spectral ordering gives an ordering for the haplotypes – that is, an ordering $i_1, i_2, \ldots, i_n$ such that $z_{i_1} \leq z_{i_2} \leq \cdots \leq z_{i_n}$ – in which similar haplotypes, pairs with a high value of $S(i,j)$, tend to be close to each other because the minimization of Eq. (3.2) is mostly affected by terms with large $S(i,j)$ and large $(z_i - z_j)^2$, so the optimum should assign close-by values of z_i, z_j for all i, j where $S(i,j)$ is high.

With similar haplotypes being clustered together, we can now assume that the SNPs that are monoallelic in different subpopulations are set together. If there are only two such subpopulations, then minimizing Eq. (3.2) results in distinct subpopulations being separated in the ordering; more specifically, the indices for the haplotypes in a subpopulation are clustered together.

A good partition $\{\mathrm{N}, \mathrm{I}\}$ of the haplotypes is then found by considering for each possible cutpoint j the partition into the components $\mathrm{N}_j = \{i \,|\, z_i < z_j\}$ and $\mathrm{I}_j = \{i \,|\, z_i \geq z_j\}$, and selecting the pair $\{\mathrm{N}_j, \mathrm{I}_j\}$ with the highest normalized bicomponent score.

Experiments on synthetic data sets showed that there were no large differences between different methods. InvCoal, which is described in Chapter 2, was used to simulate 250-kb-long segments completely covering an inversion. The modelled recombination rate was 10^{-9} and haplotype count of 120, with the other simulation parameters defined in Table 5.3. For SNP ascertainment we used

48-haplotype panel ascertainment. This is briefly described in Section 5.2.1. The simulated inversion age was 40,000 generations.

To measure the performance of the different methods, *receiver operating characteristic* (ROC) curves were used. Assume that the cumulative distribution functions of positive and negative cases separately for one method are $T(x)$ and $F(x)$, with x being a real number. A case is labelled positive if the score for the data point is higher than x. In such case, $1 - T(x)$ gives the true positive fraction or the power for the threshold x and $1 - F(x)$ gives the false positive fraction or $1 -$ specificity for the same threshold. By letting x vary, the points $(1 - F(x), 1 - T(x))$ plot the ROC curve. By fixing the power, the corresponding specificity can be read from the curve and vice versa. Generally speaking, the higher the curve is with low values of false positive fraction, the better the method performs.

In this case, the positive cases were the data sets that contained a simulated inversion whereas the negative cases did not. There were 1,000 data sets of both types. In effect, the ROC curves show how well NBS with the specific method for selecting the subpopulation division could be used to detect inversion presence in the data set.

To summarize and compare different methods, the area under the ROC curve (AUC) [123] was used. In the end, we selected spectral ordering with the informative prior. The AUC values are listed in Table 3.1. The closer the value is to one, the better the method performed. The simulated case was deliberately set as a difficult one, as the chosen recombination rate r is quite low. The value was chosen for the AUC not to be set very close to 1 for easier comparison between different methods.

In the results table 3.1 we see that k-Means and its variants, including k-Means++, generally did not perform well. This is due to the fixed way of dividing the data set in two; k-Means++ ordering is one way of overcoming this by ordering the haplotypes on the real line and then selecting the best division of the 120 possible ways. An alternative way of organizing them in this case would have been to project the haplotypes from the multidimensional space onto the line passing through the inferred centroids, but this option was not investigated further. As spectral ordering with the informative prior seemed to perform best in these tests, it was chosen as the method of splitting the haplotype population in two in the experiments.

Table 3.1: AUC-values for different methods for dividing the haplotypes in two populations and different priors for NBS.

Inv.pop. proportion (%)	k-Means	k-Means++	Spectral ordering
10	0.6427	0.6182	0.6496
20	0.9074	0.9077	0.9064
30	0.9670	0.9673	0.9666

	Spectral ordering, informative prior	k-Means++ ordering with inf. prior	
10	0.6761	0.6741	
20	0.9154	0.9135	
30	0.9706	0.9703	

3.4 Distinguishing haplotype blocks from inversions

As mentioned in Section 3.1, haplotype blocks can also fit our four-field model of inversion effects on nucleotide variability, which raises the question of how to discern inversions and haplotype blocks from each other. In haplotype blocks, recombination rates have been suppressed practically completely, while in inversions they typically have not been suppressed in homokaryotypic individuals. This is the trait we aim to measure to improve our accuracy in accurately labelling haplotype blocks apart from inversions.

Let us assume that the proportion of different karyotypes are in Hardy–Weinberg equilibrium and that only the production of inviable gametes reduces the proportion of viable recombinant gametes in a generation. With these assumptions, we can roughly bound the suppressed recombination rate per generation. Because the suppression occurs only in heterokaryotypes, the expected number of recombinations in the population can be bounded by investigating the recombinations in homokaryotypes only.

If we assume homo- and heterokaryotypes to be in the Hardy–Weinberg equilibrium and that recombinations are equally likely to be proposed regardless of the karyotype, we see that the factor by which the number of recombination is reduced is at most 0.5.

This is because regardless of the arrangement mixing proportion, homozygotes constitute at least 50% of the population. Hence, if all proposed recombinations in homozygotes are expected to result in viable gametes, then the number of recombinations per generation is at most halved within the inversion region.

This would let us believe recombinations would still be frequent in the inversion region, if the region was not a recombination cold spot prior to the inversion formation. Note that this does not mean that the rarer arrangement is affected by many recombinations.

Thus, we should be able to distinguish haplotype blocks from inversion regions by evaluating the presence of recombinations within the suggested subpopulations and between them; by the 'within' and 'between' recombination rates we refer to recombinations occurring in homokaryotypes and heterokaryotypes, respectively.

One common characteristic of haplotype blocks is the suppression of recombinations to a level greater than that in inversions. This distinction makes the estimation of recombination rates interesting: if the investigated region has few signs of recombinations in spite of potential for evidence in the contrary, then this site is more likely to be in a haplotype block than otherwise. This is shown as a smaller number of distinct haplotypes within the block.

InvCoal simulations show us, however, that the observed recombinations are mostly limited to one subpopulation, the ancestral type, at least in the scenarios InvCoal can handle moderately well, which limits the simulation to the cases where the inversion-type population is the smaller one. If we assume the inversion haplotypes to be rarer than the ancestral orientation haplotypes, then the haplotypes in the sample quickly find their most recent common ancestor due to the small number of potential parents in each generation. Hence the haplotypes in the newer population are expected to be more alike than in the other if the population is also the smaller one.

Also, because the MRCA for the inversion population is resolved usually well before the inversion event, a considerable amount of mutations are introduced within this arc in the genealogy, resulting in SNPs that are indicators for inversion-type haplotypes. It follows that in this subpopulation monomorphic SNPs that cannot be used to infer recombination rates are present in higher proportion than otherwise. Hence, the estimated recombination rates for this region

are unreliable, but could be assumed to be considerably smaller due to the rarity of inversion-type homozygotes with the assumption of inversion-type haplotypes being the rarer type.

In this sense, assuming we have correctly inferred the subpopulation division, we can take the maximum estimated recombination rate, or other statistic for estimating the frequency or presence of recombinations, of the two subpopulations separately. In haplotype blocks, we expect both inferred populations to have low recombination rates, whereas in inversion regions, we expect the ancestral type population to have higher recombination rates than in haplotype blocks. Because this is still assuming that the inversion-type population is the smaller one, this cannot be used to tell which subpopulation is the ancestral type, as the assumption might not hold in reality. The output of InvCoal should also not be given too much weight in deciding such matters due to the inaccuracies in the underlying model.

There are multiple methods proposed to estimate the recombination rate from SNP data sets [121]. For example, there are methods based on pairwise allele incompatibilities such as Φ_w [16] and methods to estimate the minimum number of recombinations explaining the SNP allele patterns observed [58].

Coalescent-based methods (e.g. [35, 56, 84]) give results frequently with the effective population size as a factor in the form of $\rho = 4N_e r$, which makes the interpretation of the results harder for the case of inversions because the effective population size might be difficult to infer in the case of inversions and their past population size. As the family trees the coalescent produces are a product of a random process with a computable likelihood, it makes sense to use the coalescent model with probabilistic methods to estimate the recombination rate.The methods to accomplish this include Markov chain Monte Carlo [73, 90] and importance sampling [35, 84]; these methods typically sample the space of coalescent trees compatible with the sample and with varying recombination and mutation rates. Because of the generation of ancestral recombination graphs, coalescent methods also require the specification of the population history in terms of effective population size. The presence of inversions makes this requirement more problematic, as they effectively decrease the effective population size by splitting the joint population in two independently coalescing subpopulations with their own

effective population sizes. Hence these methods may be unreliable for this case.

The coalescent model can be used to estimate full likelihood, which is computationally very intensive, or composite likelihoods [56, 84], which considers only pairs (or more) of SNPs at a time. In this approximation, the pairs are considered independent. Because multiple windows need to be evaluated for their recombination rates at a rapid pace, full likelihood methods are unusable. As a third way of estimating recombinations we mention R_M by Hudson and Kaplan [58]. This is an estimate for the minimum number of recombinations that have occurred in the history of the subpopulation. The estimate resolves nonoverlapping intervals that have to contain at least one recombination to explain the data and then returns the number of these intervals as the estimate.

Of the three estimates mentioned, R_M and Φ_w are as concepts perhaps closest to what is being sought: evidence of recombination within the relevant region. A problem with Φ_w is that it is used to test hypothesis of the presence of at least one recombination. This leads into handling small p-values for which it might not be easy to set a threshold, and also questions how well the statistic actually measures the desired feature. Both statistics are also dependent on the length of the interval. Another problem is that when joining together sliding windows over a genome, regions outside the interesting region, be it an inversion or a haplotype block, will add to these estimates.

Of the three statistics, R_M was chosen to be used later on. Φ_w is excluded in this thesis for the problems with interpretability and fixing the used thresholds. The third measure, ρ, is excluded due to the problem with eliminating the effect of the effective population size and hence also problems with setting the threshold. Other ways of incorporating these statistics may, however, make them more useable for the purpose of filtering out false positive data sets.

3.5 Inversion-detection algorithms

The pieces presented in this chapter can now be merged into a single algorithm for scanning a complete chromosome to detect the presence of inversion polymorphisms. The NBS-Scan algorithm is

given as Algorithm 3.1.

The NBS score has no direct interpretation as a p-value. For this reason, the threshold values in deciding which windows are considered inversion polymorphism candidates are not given a measure of statistical significance. Also, as shown in Section 5.2.4, the distribution for the score is strongly affected by the effective population size in simulations.

NBS-Scan progresses by sliding a window over the genome in fixed-width steps. In the implementation used in the rest of the thesis, the window moves in 50-kb steps.

Overlapping windows are joined to form larger regions for putative inversions. Although unlikely, it is possible that adjacent windows have strong support for different population subdivisions. By combining overlapping windows, we can, in part, avoid this by computing NBS for the combined window, thus seeing how well one subdivision can be used to explain the pattern.

Finally, to eliminate haplotype blocks as false positives, a measure for recombinations for the candidate regions can be required to exceed the threshold given as a parameter. This parameter is used in step 6 of Algorithm 3.1.

NBS-Scan was chosen to progress by basepairs instead of by SNPs because the impact of a single SNP for NBS is low in cases where the number of SNPs is reasonably high. Beside that, SNPs that are located close to each other are typically in high LD, which also means that they might mislead NBS in regions of high SNP density, if the sliding window has a constant number of SNPs. By moving the window by 50 kb at a time, these effects were partially mitigated at the cost of not having possibly more accurate inversion end-point estimates.

Let us now consider an alternative method of using NBS in detecting inversions. This approach is mostly based on Sindi and Raphael's recent method [112]; this method is henceforth called SR-method. We first review the outline of it.

In brief, SR-method considers the spaces between two pairs of SNPs as potential inversion end-points. The likelihoods of the haplotype frequencies of the SNPs next to the end-points are modelled by forming two haplotype blocks, both of which contain one end-point. The haplotype frequencies are then estimated both for the null model of no inversion present and a two-component mixture

Algorithm 3.1 NBS-Scan algorithm for listing candidate inversion regions.

Input: SNP data set, recombination measure threshold
Output: List of candidate regions

1: Divide the sequence in overlapping windows $W = (w_1, w_2, \ldots, w_m)$.
2: Remove the SNP-free parts at the ends of each window.
3: **for** $i = 1, \ldots, m$ **do**
4: Divide w_i in N and I by spectral ordering with the informative prior $\alpha = \beta = 0.1$.
5: Compute $\mathrm{NBS}(w_i|\mathrm{N}, \mathrm{I})$.
6: Compute the recombination measure for the inferred populations.
7: Ignore windows where the recombination measure was below a given threshold for the measure.
8: **end for**
9: Remove windows with NBS below the NBS threshold.
10: If any remaining windows with NBS above a fixed threshold overlap, merge them.
11: Find $\{\mathrm{N}, \mathrm{I}\}$ for each contiguous region by spectral ordering.
12: Compute NBS for the joined windows and eliminate regions with score below the NBS threshold.
13: List the remaining regions.

model where one component contains the inversions.

For interpreting the results, it is relevant to detail the process of deciding which intervals were potential inversion end-points. For real data, this is also described by Sindi and Raphael [112]. From the data sets, be they real or synthetic data sets without inversions, the entropy of haplotypes in SNP blocks with length $2L$, where $L = 3, \ldots, 15$, is computed. From the resulting histogram, the value of entropy that marked the limit of top 10% was stored for each value of L.

The next step is to create an empirical distribution for estimating the p-value for a pair of putative inversion end-points. Each gap between two adjacent SNPs was investigated by using a specific procedure. Starting from $L = 3$, the L SNPs to both left and right of the gap were used to form a block and its entropy was computed.

If the entropy was in the top 10% of the previous simulations, the gap was marked for further processing as a potential end-point. Otherwise, L was increased by one and the evaluation was repeated. This continued until L was 15 or the gap was marked for further processing.

Next, each pair of potential end-points where the distance between them was over 200 kb and the block configurations did not overlap were used to compute the likelihood ratio by the EM-algorithm for the likelihoods of the two haplotype block models of the same blocks, one with only one component (the case of no inversion present) and a mixture of two components (haplotypes with and without inversions). The likelihood ratio test is further discussed in Section 4.5. Sindi and Raphael [112] note that the χ^2 distribution in this case is a poor approximation of the actual distribution, so they use an empirical distribution to compute the p-values instead.

To form this distribution, the distance between the end-points and the degrees of freedom were stored in a table alongside the likelihood ratio. After the experiments, the values in this table were used to evaluate the empirical p-value for the putative inversion in the actual simulations.

We can now specify a hybrid method that attempts to combine NBS with SR-method. It utilizes NBS in deciding which gaps between SNPs are potential inversion end-points. By computing NBS from two 50-SNP windows, both in different directions, one minus the absolute difference between these is used to multiply the p-value of the entropy computed for the haplotype frequencies. The potential end-points are chosen based on this product rather than the entropy p-value alone. Note that if the computed NBS values are the same and the threshold for including the SNP gap is 10%, the hybrid method performs like SR-method.

These inversion-detection methods are experimented on in Section 5.2.3.

3.6 Discussion

This chapter presented a scoring criterion, NBS, for detecting the presence of common inversion polymorphisms from dense SNP data

sets and an algorithm that utilizes it.

NBS-Scan detects inversions as the low recombination rate between two subpopulations. If the recombination rate within the subpopulations is high, then NBS-Scan can be expected to perform well. However, if the recombination rate is overall much lower, this is not detected as different from an inversion, which results in numerous false positives in spite of a recombination measure threshold. These claims are investigated in Sections 5.2.3 and 5.2.4 in both synthetic and real data sets.

It is well-known that recombination rates vary across the human genome. This makes accurate detection of the presence of inversions by using NBS more difficult, as regions with a low rate of recombination or relatively few SNPs can increase the number of false positives.

As mentioned, in regions of low recombination rate the SNPs are in high LD with each other. Because this can produce windows with a high NBS, it is necessary to specify the window to be wide enough for including a region sufficiently long in terms of genetic distance. This, unfortunately, has the drawback of making the scoring more insensitive towards short inversions. Hence a balance between these two must be found.

Another aspect is that the scoring prefers regions that coincide with the yin yang -haplotype pattern described by Zhang et al. [146]. In this pattern the population is divided into haplotype blocks, two of which have archetypes that are complementary to each other. It is possible that NBS-Scan proposes such regions to be inversions.

NBS-Scan presented attempts to discover a region slightly larger than the actual inversion region. More accurate methods for estimating the end-points of the actual inversion are not easy to devise. Different heuristics have been experimented on to improve the accuracy of the estimated end-points, but a completely automated method for that was not successfully produced.

One approach that was investigated for estimating the end-points more accurately, is to consider how unlikely it is for each SNP individually have its alleles split in the way they are in the inferred subpopulation division. As the distance from the inversion increases, recombinations break the division present. One putative scoring per SNP would be the number of data matrix element flips required for

the SNP to be monomorphic in at least one inferred population. By assigning haplotype miscall a probability and assuming phasing errors do not exist, it is possible to compute the likelihood that the data actually is monomorphic. It is then possible to select a base level of significance, assuming the SNP alleles were independently and randomly split between the two inferred arrangements, and then find the interval where the significance was the highest. Unfortunately, this approach likely requires the manual selection of the base level, as a reasonable bipartition eliminates the independence of allele division into two subsets.

The methods of Bansal et al. [9] and Sindi and Raphael [112] utilize different signal than our approach, as they focus on the signal in the LD patterns near the inversion end-points. This raises the question which signal is clearer and in which conditions. Some experimental results are presented in Section 5.2.3.

CHAPTER 4

Detecting deletions

Deletion polymorphisms are a particular variant of copy number variants (CNVs). In deletions, one or multiple copies of a chromosome segment are present in some people and missing in others. We examine the signal that deletion polymorphisms left in the genotype data sets. We review the framework of an Expectation-Maximization [27] algorithm for estimating haplotype frequencies. This algorithm is then shown to work also for detecting the presence of deletion polymorphisms in genotype data sets collected either as trios or unrelated individuals, the latter case being a novel finding. Two methods that are computationally more efficient than the trivial implementation are presented. The difference to previously existing methods is in the improved time complexity.

Finally, we discuss the problems in using the likelihood ratio test for the significance of the detected deletions.

4.1 Biological signal and related work

A common method of detecting which genotypes a person has is to use DNA microarrays to detect which alleles of a SNP are present in the sample. If both alleles are present, the SNP is considered heterozygous. If only one allele is present, the SNP is then called homozygous.

Because only the presence of the allele in either of the two strands is detected, a hemizygous deletion, i.e., one copy of the chromosome has the deletion, is observed as a series of homozygous alleles or

null genotypes, where neither allele is found. To simplify the case somewhat, if the deletion is homozygous, i.e. both strands where the SNPs are located are deleted, the SNPs are read as missing alleles or no calls. This case is shown in in the rightmost example in Figure 4.1.

If there are several hemizygous deletions present in the sample, this results in more genotypes called homozygous than would be expected according to Hardy–Weinberg equilibrium. This departure from the equilibrium can then be detected.

Another detectable signal in trio data are Mendelian inconsistencies. In trios, the child inherits one haplotype from both parents. If the parents are homozygous, then the child's haplotype should be completely determined assuming no errors in measuring the genotypes and no mutation in the child.

Let us consider the case of a hemizygous deletion in the mother and assume that the child inherits the deletion haplotype. The child, assuming that the haplotype inherited from the father is not a deletion haplotype, now reads as a homozygote of the other (here the father's) parental haplotype. If the paternal inherited haplotype differs from the maternal non-inherited haplotype, this is read as a Mendelian inconsistency. This case is depicted in Figure 4.1 (b).

Similarly to how the alleles not being in complete Hardy–Weinberg equilibrium can follow from pure chance, Mendelian inconsistencies can result from genotyping errors. Hence the discovery of deletion polymorphisms is not as straightforward as merely finding all trios and SNPs with such inconsistencies.

Altogether, these signals can be detected by various means from the SNP data. Kohler and Cutler [68] examine each SNP separately before joining putative deleted SNPs into windows. This approach creates estimates of the underlying haplotypes and error rates in the genotyping process. These estimates are then used to decide the presence and the limits of the deletions in a probabilistic framework.

McCarroll et al. [81] look for nearby SNPs with similar failure profiles. These failure profiles are binary vectors for each SNP and depict patterns of null genotypes, Mendelian inconsistencies or the combination of these two. If genotyping errors that are not due to structural variants are expected to be independent, nearby SNPs with similar failure profiles are possibly due to structural variants. Therefore, these failure profiles are clustered and the significance

level for the detected pattern in the SNPs is evaluated. McCarroll et al. also use deviations from the Hardy–Weinberg equilibrium to combine the SNP clusters inferred from the failure profiles.

Likewise, Conrad et al. [20] use evidence directly countable from trio data to detect the presence of deletions. Each detected trio of genotypes is evaluated whether it contains Mendelian errors supportive of deletions, for instance, a homozygous child with a homozygous parent of a different allele. If there are sufficiently many such inconsistencies, a deletion is estimated to be present.

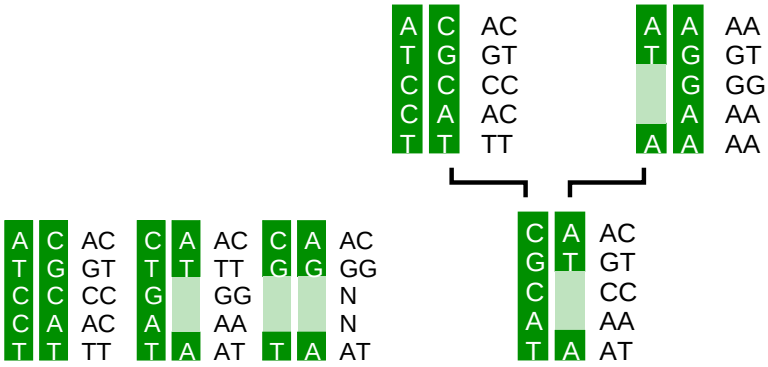
Franke et al. [40] use oligonucleotide arrays to detect deletions from the genotyping experiment data. In a way, this work preempts the need for methods that depend solely on the measured genotypes by utilizing direct probe intensity readings before the data is discretized into genotype calls.

Of particular interest to this thesis is the work of Corona et al. [22]. They detect the presence of deletions by using haplotype frequency estimation methodology to evaluate how well the genotype data in a specific window can be explained first by all possible non-deleted haplotypes within that window, and then by adding a deletion haplotype to the potential set of true haplotypes. The deletion status is then inferred based on the difference in the data likelihoods in these two models.

4.2 Estimating haplotype frequencies

For detecting the presence of a deletion, this thesis adapts the approach presented by Corona et al. [22]. The estimation of haplotype frequencies has been a widely studied problem under a number of different cases [33, 147].

The EM algorithm used by Corona et al. modelled genotyping under errors. The presented algorithm had time complexity of $O(k^4)$ where k is the number of different haplotypes that may have a positive frequency in the population. If all possible haplotypes are being considered, then in a window of m SNPs k equals 2^m. This made the algorithm infeasible in practice for long windows. Instead, they used the haplotyping program HAP [45] to discover initial haplotype frequencies without deletion, and then scaled the deletion frequencies with the EM-algorithm. Each iteration now



(a) In unrelated individuals a deletion may cause clusters of null genotypes (N) and an excess of homozygotes (a hemizygous deletion is incorrectly typed as a homozygous genotype of the undeleted allele)

(b) In mother–father–child trios a transmitted deletion may, additionally, cause a cluster of apparent Mendelian inconsistencies: depicted is a situation where at one SNP within the deleted region a child is typed as a homozygote (here CC), even though the parents are typed as different homozygotes (here CC and GG; more generally we refer to the different alleles as 0 and 1); the genotypes at the second SNP within the deleted region, however, are consistent with the Mendelian laws.

Figure 4.1: Idealized footprint of a deletion in SNP genotype data, in the absence of genotyping error. On green background are the true haplotypes; the observed genotype is next to them on white.

takes time $O(k^3)$. Because this is still inefficient, they focus only on regions where they believe their method can detect deletions.

The EM algorithm for estimating the haplotype frequencies is here made computationally feasible by showing how the likelihood for a trio can be computed in $O(k^2)$ time (per EM iteration). Furthermore, a further adjustment to the algorithm results in a runtime of $O(m2^m)$ for m markers. This means that when k is of the order of 2^m, the running time becomes $O(k \log k)$. This remains computationally feasible when m is fairly small (at most 10). If k is below $O(2^m)$, the time complexity of the algorithm still will be $O(m2^m)$.

This method can be applied to trios and unrelated individuals,

thus extending the applicability of the approach to a large class of study designs.

Let us focus on a window of m SNPs. In total, there are 2^m non-deletion haplotype possibilities with two choices, 0 or 1, at each position.

We first limit the focus into $k \leq 2^m$ distinct haplotypes and mark them $H_1, H_2, \ldots, H_k$, each being a sequence in $\{0,1\}^m$. The value of each SNP s in haplotype H_i is denoted by $H_i^s \in \{0,1\}$. Breaking the binary form, we write the deletion haplotype as $H_0^s = \mathrm{D}$ for all s. In this model, this is the only haplotype that contains deleted SNPs and all of the SNPs are deleted. To each haplotype we affix their population frequencies $f_0, f_1, f_2, \ldots, f_k$ so that $f_0 + f_1 + f_2 + \cdots + f_k = 1$.

A pair of haplotypes H_i and H_j determine a genotype $G(H_i, H_j) = G(H_j, H_i) = G = G^1 G^2 \cdots G^m$. Each single-SNP genotype G^s has six possible values: homozygous 00 and 11, heterozygous 01, hemizygous deletions 0D and 1D and homozygous deletion DD. The order of the haplotypes is irrelevant; hence the hetero- and hemizygous cases both represent two different haplotype assignments. For instance, if $H_i^s = \mathrm{D}$ and $H_j^s = 1$, then $G^s = 1\mathrm{D}$.

In the data sets, Mendelian inconsistencies may be observed as, for example, the child apparently inheriting an impossible haplotype from one or both of the parents, but these can occur not only in the presence of deletions but because of errors in measuring the genotypes. For this reason the observed genotype is modelled here as a product of two 'true' haplotypes after applying a probabilistic error mask. We denote the observed genotype with $\hat{G}$ and the four possible values it can have by 0, 1, 2 and N. Values 0 and 1 represent corresponding homozygotes, 2 the heterozygote and N *no call*, null genotype or missing data. Note that also hemizygous deletions are observed as homozygotes (Figure 4.1).

Whereas the deletion haplotypes deterministically modify the observed genotypes, the genotyping error mask changes the underlying true genotypes into the observed ones with specific probabilities. Let the true genotype be G^s and the observed genotype $\hat{G}^s$. We now denote the probability of observing $\hat{G}^s$ instead of G^s by $\varepsilon_s(G^s, \hat{G}^s)$. With the simplifying assumption of the errors being independent, the probability of observing the genotype sequence $\hat{G}$

with real genotype sequence G becomes

$$\varepsilon(G, \hat{G}) = \prod_{s=1}^{m} \varepsilon_s(G^s, \hat{G}^s).$$

With the standard assumptions of random mating and Hardy–Weinberg equilibrium, summing over all true genotypes results in the probability of observing $\hat{G}$,

$$L(\hat{G}) = \sum_{i=1}^{k} \sum_{j=1}^{k} f_i \, f_j \, \varepsilon(G(H_i, H_j), \hat{G}) \,. \tag{4.1}$$

4.2.1 Expectation-Maximization algorithm

The Expectation-Maximization algorithm framework [27] has been successfully applied to a variety of different problems to find parameter values to maximize the likelihood of the parameters, which is defined as the probability of the observed data given the parameters: $L(f) = \Pr(\mathcal{D}; f)$. One particular application for it is the estimation of haplotype frequencies for both unrelated individuals (e.g. [33, 147]) and trios (e.g. [147]). By using simulated data, it has been shown to produce accurate estimates of the haplotype frequencies [34].

We now briefly review the features of the framework; we call it a framework, as the form of the actual algorithm depends on the data likelihood formula. These derived algorithms are iterative algorithms that start from one parameter configuration, after which they deterministically update the model parameters. The updates monotonically increase the likelihood. These iterations are continued until the likelihood appears to have converged, i.e., the likelihood no longer increases considerably between iterations. As a result, the algorithms are guaranteed to find a local optimum of the likelihood. Multiple random starting points can then be used to increase the possibility of finding the global optimum.

The EM-algorithm complements the observed data, $\mathcal{D}$, with unobserved (missing) data $\mathcal{D}^M$. Together these are called the complete data. In the case of haplotype frequency estimation, the observed data are the observed genotypes, and the missing data are the true underlying haplotypes. We then account for all possible values of $\mathcal{D}^M$ by marginalizing over it.

The algorithm itself consists of two steps that are repeated until the likelihood converges: the Expectation or E-step and the Maximization or M-step. In E-step, the conditional expectation of the missing data is computed given the observed data and the estimated parameters. In the M-step, the new parameter values that maximize the expected likelihood of the complete data are computed (where the expectation is over the distribution of the missing values in the E-step).

Let us denote the parameters we wish to optimize as $f^{(1)}$ and the parameters from the previous iteration as f. The conditional expectation of complete data log-likelihood $\log L_c$ is denoted

$$\begin{aligned} Q(f^{(1)}|f) &= \mathrm{E}_{\Pr(\mathcal{D}^M|\mathcal{D},f)}(\log(L_c(f^{(1)}|\mathcal{D}^M,\mathcal{D}))\,|\,\mathcal{D}) \\ &= \int \log L_c(f^{(1)}|\mathcal{D}^M,\mathcal{D})\Pr(\mathcal{D}^M|\mathcal{D},f)\mathrm{d}\mathcal{D}^M \end{aligned}$$

with $\Pr(\mathcal{D}^M\,|\,\mathcal{D},f)$ being the probability density function of missing data given the parameters f and the observed data. In the case of haplotype frequency estimation, f is the vector of estimated haplotype frequencies and the integration over all $\mathcal{D}^M$ becomes summation over all haplotype combinations compatible with the observations.

To properly perform the M-step, it is necessary to find the value of $f^{(1)}$ that maximizes $Q(f^{(1)}|f)$. Note that the previous parameter estimates f affect the distribution $\Pr(\mathcal{D}^M\,|\,\mathcal{D})$ only; on the other hand, the next iteration parameters, now considered random variables, are present only in the complete data likelihood function.

This maximization step is dependent on the actual formulation of the likelihood. In later chapters, we consider cases where the data is either trios or unrelated individuals.

4.2.2 Error models

The definition of genotype likelihood in Eq. (4.1) leaves us with the task of specifying the error probabilities. We present an error model parametrized by two error rates shared by all SNPs. The miscall rate τ represents the possibility of observing a haplotype allele different from the true one and the no call rate δ represents the possibility of reading the genotype as missing. We assume these errors occur independently for each SNP and haplotype and that

a deletion haplotype is never read as another allele. With these parameters and assumptions we form a six-by-four table of probabilities as seen in Table 4.1. For instance, if the true genotype is 00, we observe genotype 1 by miscalling both haplotypes independently with joint probability τ^2 and not reading it as a null genotype, adding the factor of $(1 - \delta)$. Similarly, if the true genotype is 01, we observe it as a heterozygous 2 by either calling both haplotypes correctly with probability $(1 - \tau)^2(1 - \delta)$ or both incorrectly (probability $\tau^2(1 - \delta)$).

The model specified above is only one parametrized model of what the error model can represent. It is also possible to use SNP-specific values for τ and δ, and also permit detecting an allele present also with homozygous deletions.

Indeed, there are other proposed models of varying complexity. For instance, Kohler and Cutler [68] proposed a model with six parameters: the probability of miscalling a homozygote as a heterozygote, the probability of miscalling a heterozygote as a heterozygote, the probability of miscalling a homozygote as the other homozygote and three separate missing data rates for different real genotypes.

By comparison, Corona et al. [22] parametrized their model with miscall probabilities and the no call probability which were estimated from data, but directly substituted the values in Table 4.1 with these probabilities rather than harmonizing the probabilities by means of underlying true parameters, τ and δ in our case. This is also close to the model used in this thesis as both share the same missing data rate parametrization. The difference hence lies in the specifics of how miscalls are modelled.

4.3 Efficient implementation

When investigating whole-genome data sets, it is preferable that the detection algorithms are efficient yet do not achieve the speed at the cost of accuracy. In this section, we consider the cases of data sets of trios and data sets of unrelated individuals separately. For both cases, we review the EM-algorithm derived by Zou and Zhao [147] for estimating haplotype frequencies under genotyping errors, but also present efficient methods for performing the M-step.

Table 4.1: True-Genotype/Observed-Genotype Probability Matrix

		Observed genotype			
		0	1	2	N
True genotype	00	$(1-\tau)^2(1-\delta)$	$\tau^2(1-\delta)$	$2\tau(1-\tau)(1-\delta)$	δ
	11	$\tau^2(1-\delta)$	$(1-\tau)^2(1-\delta)$	$2\tau(1-\tau)(1-\delta)$	δ
	01	$(1-\tau)\tau(1-\delta)$	$\tau(1-\tau)(1-\delta)$	$((1-\tau)^2+\tau^2)(1-\delta)$	δ
	0D	$(1-\tau)(1-\delta)$	$\tau(1-\delta)$	0	δ
	1D	$\tau(1-\delta)$	$(1-\tau)(1-\delta)$	0	δ
	DD	0	0	0	1

4.3.1 Data model

We assume that the trios or individuals have been independently sampled. Thus, we can write the likelihood term (probability of the data) for n trios $\hat{T}_1, \hat{T}_2, \ldots, \hat{T}_n$ by assuming independence as

$$L(\hat{T}_1, \hat{T}_2, \ldots, \hat{T}_n) = L(\hat{T}_1)\, L(\hat{T}_2) \cdots L(\hat{T}_n)\,.$$

Although likelihood is technically considered a function of model parameters, for convenience we instead point out the data for which the likelihood is computed. For example, $L(\hat{T}_1)$ denotes a likelihood function corresponding to having observed $\hat{T}_1$, and so on.

The likelihood for data from a mother–father–child trio takes into account the Mendelian dependencies among the trio's genotypes. Let $\hat{M}$, $\hat{F}$, and $\hat{C}$ be the observed genotypes for the mother, the father and the child, respectively, for one trio. Assuming that the parents each transmit one of their two haplotypes as such, without recombination, the underlying haplotypes of the trio can be specified by four haplotypes: the mother's transmitted and untransmitted haplotype, say H_i and $H_{i'}$, respectively, and the father's transmitted and untransmitted haplotype, say H_j and $H_{j'}$, respectively. The true genotypes of the mother, the father, and the child are then $M = G(H_i, H_{i'})$, $F = G(H_j, H_{j'})$, and $C = G(H_i, H_j)$, respectively. The likelihood for the observations $\hat{T} = (\hat{M}, \hat{F}, \hat{C})$ is obtained by summing over the four unobserved haplotypes,

$$L(\hat{T}) = \sum_{i=0}^{k} \sum_{i'=0}^{k} \sum_{j=0}^{k} \sum_{j'=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \varepsilon(M, \hat{M})\, \varepsilon(F, \hat{F})\, \varepsilon(C, \hat{C})\,,$$

where f is the vector of relative haplotype frequencies for k haplotypes and f_i, $f_{i'}$, f_j and $f_{j'}$ are elements of that vector telling the frequencies of haplotypes H_i, $H_{i'}$, H_j and $H_{j'}$, respectively. Here, again, we assume that the haplotypes are paired independently.

Above, with several independent trios the likelihood (probability of observed data) became the product of the individual probabilities of each trio; similarly, for n independently sampled individuals $\hat{G}_1, \ldots, \hat{G}_n$ the likelihood becomes

$$L(\hat{G}_1, \hat{G}_2, \ldots, \hat{G}_n) = L(\hat{G}_1)\, L(\hat{G}_2) \cdots L(\hat{G}_n)\,.$$

The probability of the observed genotype of one individual is shown in Eq. (4.1).

4.3.2 Trio datasets

We now show how to derive the EM-algorithm for genotyping under errors in trios. This has been derived by Zou and Zhao [147], but we review it here following our notation.

For the algorithm, we select the unknown true haplotypes as the missing data and the haplotype frequencies f as the parameters. We denote $\Pr(\hat{T}_t|H_i, H_{i'}, H_j, H_{j'})$ as the probability of observing trio $\hat{T}_t$ with mother's, father's and child's haplotypes denoted as above; more formally,

$$\begin{aligned}\Pr(\hat{T}_t|H_i, H_{i'}, H_j, H_{j'}) &= \varepsilon(G(H_i, H_{i'}), \hat{M}_t)\, \varepsilon(G(H_j, H_{j'}), \hat{F}_t) \\ &\quad \times \varepsilon(G(H_i, H_j), \hat{C}_t).\end{aligned}$$

Now, we can write the expected log-probability of the complete data given the previous iteration's frequencies f as

$$\begin{aligned}Q(f^{(1)}|f) &= \sum_{t=1}^{n}\sum_{i=0}^{k}\sum_{i'=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k}\Big(\log\Pr(\hat{T}_t|H_i, H_{i'}, H_j, H_{j'}) \\ &\quad + \log\Pr(H_i, H_{i'}, H_j, H_{j'})\Big) \\ &\quad \times\Pr(H_i, H'_i, H_j, H'_j|\hat{T}_t, f) \\ &= \sum_{t=1}^{n}\sum_{i=0}^{k}\sum_{i'=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k}\Big(\log\Pr(\hat{T}_t|H_i, H_{i'}, H_j, H_{j'}) \\ &\quad + \log\left[f_i^{(1)} f_{i'}^{(1)} f_j^{(1)} f_{j'}^{(1)}\right]\Big) \times \Psi_t(i, i', j, j'),\end{aligned}$$

where

$$\begin{aligned}\Psi_t(i,i',j,j') &= \Pr(H_i, H'_i, H_j, H'_j|\hat{T}_t, f)\\ &= \Pr(\hat{T}_t|H_i, H_{i'}, H_j, H_{j'}) f_i f_{i'} f_j f_{j'}\\ &\quad\times\left(\sum_{l_1=0}^{k}\sum_{l_2=0}^{k}\sum_{l_3=0}^{k}\sum_{l_4=0}^{k}\Pr(\hat{T}_t|H_{l_1}, H_{l_2}, H_{l_3}, H_{l_4})\right.\\ &\quad\left.\times f_{l_1} f_{l_2} f_{l_3} f_{l_4}\right)^{-1} \qquad (4.2)\end{aligned}$$

is the probability that the underlying true haplotypes are $H_i, H_{i'}, H_j$ and $H_{j'}$ if we observe trio $\hat{T}_t$.

The divisor in Eq. (4.2) is a constant for the trio and the likelihood of the trio under the previous iteration haplotype frequencies; we denote this sum $L_0(T_t)$.

With the additional constraint of $\sum_{h=0}^{k} f_h^{(1)} = 1$, we can use Lagrange multipliers to see that $Q(f^{(1)}|f)$ is maximized with respect to $f^{(1)}$ when we give the parameters the following values:

$$\begin{aligned}f_h^{(1)} &= \frac{a_h}{\sum_{h'=0}^{k} a_{h'}},\\ a_h &= \sum_{t=1}^{n}\sum_{l_1=0}^{k}\sum_{l_2=0}^{k}\sum_{l_3=0}^{k}\Big(\Psi_t(h,l_1,l_2,l_3) + \Psi_t(l_1,h,l_2,l_3) +\\ &\quad \Psi_t(l_1,l_2,h,l_3) + \Psi_t(l_1,l_2,l_3,h)\Big)\\ &= \sum_{t=1}^{n}\left(\sum_{i'=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k}\Psi_t(h,i',j,j') + \sum_{i=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k}\Psi_t(i,h,j,j')\right.\\ &\quad \left.+\sum_{i=0}^{k}\sum_{i'=0}^{k}\sum_{j'=0}^{k}\Psi_t(i,i',h,j') + \sum_{i=0}^{k}\sum_{i'=0}^{k}\sum_{j=0}^{k}\Psi_t(i,i',j,h)\right).\end{aligned}$$

The last form is of particular interest. Each of the four inner sum triplets represent the probability of one of the four haplotypes being fixed as haplotype H_h given the haplotype frequencies f. Let us write the joint probability of observing a particular trio and the maternal inherited haplotype being H_i for that trio as

$$I_i(\hat{T}) = \sum_{i'=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \varepsilon(M,\hat{M})\, \varepsilon(F,\hat{F})\, \varepsilon(C,\hat{C}).$$

Similarly, we can write the joint probability for observing the trio and the maternal uninherited haplotype being $H_{i'}$ as

$$I'_{i'}(\hat{T}) = \sum_{i=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \varepsilon(M,\hat{M})\, \varepsilon(F,\hat{F})\, \varepsilon(C,\hat{C}).$$

Analogously, we define $J_j(\hat{T})$ and $J'_{j'}(\hat{T})$ for paternal inherited and uninherited haplotypes. With these, we can rewrite

$$a_h = \sum_{t=1}^{n} \frac{I_h(\hat{T}_t) + I'_h(\hat{T}_t) + J'_h(\hat{T}_t) + J_h(\hat{T}_t)}{L_0(T_t)},$$

where $L_0(T_t)$ is the denominator in Eq. (4.2).

We next look at how the joint probabilities (which could also be called augmented likelihoods) $I_h(\hat{T})$ and $I'_h(\hat{T})$ in the EM algorithm can be computed for all $h = 0, 1, \ldots, k$ in $O(k^2)$ total time. The analogous terms $J_h(\hat{T})$ and $J'_h(\hat{T})$ are computed in the same way. Note also that the total likelihood $L_0(\hat{T})$ can be computed easily by $L_0(\hat{T}) = \sum_h I_h(\hat{T})$ in $O(k)$ time once the I_h values have been computed first.

A key observation is that the joint probability expressions are sums of products where each factor in the product depends on at most two of the four haplotypes. Recall that $M = G(H_i, H_{i'})$, $F = G(H_j, H_{j'})$, and $C = G(H_i, H_j)$, and denote

$$\alpha_{ii'} = \varepsilon(M,\hat{M})\,, \quad \beta_{jj'} = \varepsilon(F,\hat{F})\,, \quad \gamma_{ij} = \varepsilon(C,\hat{C})\,;$$

note that these are symmetric matrices, as the function mapping the haplotype pairs to genotypes does not distinguish between the first and the second parameter. For each $i = 0, 1, \ldots, k$ we decompose the sum for $I_i(\hat{T})$ as

$$\begin{aligned} I_i(\hat{T}) &= \sum_{i'=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \alpha_{ii'}\, \beta_{jj'}\, \gamma_{ij} \\ &= f_i\Big(\sum_{i'=0}^{k} f_{i'}\alpha_{ii'}\Big)\Big(\sum_{j=0}^{k} f_j\gamma_{ij}\Big(\sum_{j'=0}^{k} f_{j'}\,\beta_{jj'}\Big)\Big)\,. \end{aligned}$$

Accordingly, we first compute $\tilde{\beta}_j := \sum_{j'} f_{j'}\,\beta_{jj'}$ for all j, then $\tilde{\gamma}_i := \sum_j f_j\gamma_{ij}\tilde{\beta}_j$ and $\tilde{\alpha}_i := \sum_{i'} f_{i'}\alpha_{ii'}$ for all i. Each step takes $O(k^2)$

time. Finally, we obtain $I_i(\hat{T}) = f_i\tilde{\alpha}_i\tilde{\gamma}_i$ for all i, thus taking $O(k)$ additional time, given the precomputed $\tilde{\alpha}_i$ and $\tilde{\gamma}_i$.

Similarly, for each $i' = 0, 1, \ldots, k$ we write $I'_{i'}(\hat{T})$ as

$$\begin{aligned} I'_{i'}(\hat{T}) &= \sum_{i=0}^{k}\sum_{j=0}^{k}\sum_{j'=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \alpha_{ii'}\, \beta_{jj'}\, \gamma_{ij} \\ &= f_{i'}\left(\sum_{i=0}^{k} f_i\alpha_{ii'}\left(\sum_{j=0}^{k} f_j\gamma_{ij}\Big(\sum_{j'=0}^{k} f_{j'}\,\beta_{jj'}\Big)\right)\right). \end{aligned}$$

Then we compute $I'_{i'}(\hat{T}) = f_{i'}\big(\sum_i f_i\alpha_{ii'}\tilde{\gamma}_i\big)$ for all i' in $O(k^2)$ total time.

Again, for each $j = 0, 1, \ldots, k$ and $j' = 0, 1, \ldots, k$ we compute $J_j(\hat{T})$ and $J'_{j'}(\hat{T})$ as

$$\begin{aligned} J_j(\hat{T}) &= \sum_{i=0}^{k}\sum_{i'=0}^{k}\sum_{j'=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \alpha_{ii'}\, \beta_{jj'}\, \gamma_{ij} \\ &= f_j\Big(\sum_{j'=0}^{k} f_{j'}\beta_{jj'}\Big)\Big(\sum_{i=0}^{k} f_i\gamma_{ij}\Big(\sum_{i'=0}^{k} f_{i'}\,\alpha_{ii'}\Big)\Big) \end{aligned}$$

and

$$\begin{aligned} J'_{j'}(\hat{T}) &= \sum_{i=0}^{k}\sum_{i'=0}^{k}\sum_{j=0}^{k} f_i\, f_{i'}\, f_j\, f_{j'}\, \alpha_{ii'}\, \beta_{jj'}\, \gamma_{ij} \\ &= f_{j'}\left(\sum_{j=0}^{k} f_j\beta_{jj'}\left(\sum_{i=0}^{k} f_i\gamma_{ij}\Big(\sum_{i'=0}^{k} f_{i'}\,\alpha_{ii'}\Big)\right)\right). \end{aligned}$$

We note that the above algorithm can be viewed as an instantiation of the generic variable elimination method, also called the generalized distributive law or the sum–product algorithm [2, 71, 116],[69, Ch. 3].

Also, $I_0(T)$ can be considered the unnormalized likelihood of the maternal inherited haplotype being the deletion haplotype (recall that we used index 0 to denote the deletion haplotype and indices $1, \ldots, k$ to denote other haplotypes). Analogous interpretations apply to $I'_0(T)$, $J_0(T)$ and $J'_0(T)$. We can therefore estimate which individuals in the data set carry homozygous or heterozygous deletions.

If we consider all the possible haplotypes of a window of moderate size (ca 6 to 8), the time complexity is too high for practical purposes. The complexity arises from the computation of values $\tilde{\alpha_i}, \tilde{\beta_i}$ and $\tilde{\gamma_i}$ for all $i = 1, \ldots, k$.

Each of these can be viewed as the result of the multiplication of one $(k+1) \times (k+1)$ matrix times a vector of $k+1$ elements. Naively done, the computation of each element takes $O(mk)$ time, which now stands for $O(m2^m)$. The factor m is due to the m factors contributing to the genotype observation probability. The computation of the whole result vector is hence $O(2^m 2^m m) = O(4^m m)$.

The application of Yates' algorithm [145], also treated by Knuth [67, Section 4.6.4] and Koivisto [69, Ch. 3], can improve this time to $O(k \log k)$, which corresponds to $O(m2^m)$.

Assume we have a function $q : \{0,1\}^m \to \mathbb{R}$, with the binary vector $x \mapsto q(x) = \sum_{y \in \{0,1\}^m} g_{x,y} v_y$, where $g_{x,y}$ can be factorized as $\prod_{i=1}^{n} g^i(x^i, y^i)$; x^i and y^i refer to the ith element in the respective vectors. In such case, we can compute q for all different x in time $O(m2^m)$ by Yates' algorithm, described as Algorithm 4.1.

Algorithm 4.1 An instance of Yates' algorithm in pseudocode.

Input: Factors $g^i : \{0,1\} \times \{0,1\} \to \mathbb{R}$, $t^0(y) = v_y$ for all $y \in \{0,1\}^m$.

Output: Values of t_m, which are equal to $q(x)$ for all $x \in \{0,1\}^m$.

```
for i = 1, ..., m do
    for y ∈ {0,1}^m do
        a ← y^1 ... y^{i-1} 0 y^{i+1} ... y^m
        b ← y^1 ... y^{i-1} 1 y^{i+1} ... y^m
        t_i(y) ← g^i(y^i, 0) t_{i-1}(a) + g^i(y^i, 1) t_{i-1}(b)
    end for
end for
Return t_m.
```

Let us now consider the computation of $\tilde{\gamma}$ by using Yates' algorithm. If we write $v_j \equiv f_j \tilde{\beta_j}$, then

$$\tilde{\gamma_i} := \sum_{j=0}^{k} v_j \gamma_{ij} \quad \text{for } i = 0, 1, \ldots, k.$$

We assume that $\tilde{\beta_j}$ has been precomputed.

We also note that γ_{ij}, β_{ij} and α_{ij} are of form

$$\prod_{s=1}^{m} \varepsilon(G(H_i^s, H_j^s), \hat{G}^s)\,;$$

in particular,

$$\gamma_{ij} = \prod_{s=1}^{m} \varepsilon(G(H_i^s, H_j^s), \hat{C}^s).$$

For brevity, we write $\gamma_{ij}^s = \varepsilon(G(H_i^s, H_j^s), \hat{C}^s)$; these correspond to $g_{x,y}^i$ in the above description of Yates' algorithm.

To make Yates' algorithm applicable, we separate the terms involving a deletion haplotype H_0; hence, we have

$$\begin{aligned} \tilde{\gamma}_0 &:= \sum_{j=0}^{k} v_j \gamma_{0j}\,, \\ \tilde{\gamma}_i &:= v_0 \gamma_{i0} + \sum_{j=1}^{k} v_j \gamma_{ij} \quad \text{for } i = 1, 2, \ldots, k\,. \end{aligned} \tag{4.3}$$

Since the summation in Eq. (4.3) over j goes over all possible alleles for each position $1, \ldots, m$, it can now be decomposed into m nested sums over $j_1, j_2, \ldots, j_m$ which index the allele values of H_j in each individual position; similarly, we will index the allele values of H_i by $i_1, i_2, \ldots, i_m$. For all $i = 1, 2, \ldots, k$ we have

$$\begin{aligned} \sum_{j=1}^{k} v_j \gamma_{ij} &= \sum_{j_1=0}^{1} \sum_{j_2=0}^{1} \cdots \sum_{j_m=0}^{1} \gamma_{i_1 j_i}^1 \gamma_{i_2 j_2}^2 \cdots \gamma_{i_m j_m}^m \, v_{j_1 j_2 \cdots j_m}\,, \\ &= \sum_{j_m=0}^{1} \gamma_{i_m j_m}^m \left(\cdots \left(\sum_{j_2=0}^{1} \gamma_{i_2 j_2}^2 \left(\sum_{j_1=0}^{1} \gamma_{i_1 j_1}^1 v_{j_1 j_2 \cdots j_m} \right) \right) \cdots \right), \end{aligned}$$

where $\gamma_{i_1 j_1}^1$ denotes the value of $\gamma_{ij}^s = \varepsilon(G(H_i^s, H_j^s), \hat{C}^s)$ when $s = 1$ and the alleles at H_i^s and H_j^s have values i_1 and j_1, respectively, and similarly for $\gamma_{i_2,j_2}^2, \ldots, \gamma_{i_m,j_m}^m$. The values of $j_\cdot$ are limited to the possible values the haplotype can get in non-deletion haplotypes. As such, the summations cover all possible haplotypes of m SNPs. Note that $v_{j_1 \cdots j_m}$ is merely an alternate notation for v_j to show its dependence on all the allele values $j_1, \ldots, j_m$ which constitute the haplotype choice j.

We note that this form is of the form required for Yates' algorithm to be applicable. To conclude, we can compute $\tilde{\gamma}$ in $O(m2^m)$ or $O(k \log k)$ time.

The handling of $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ is very similar: we decompose the sum into m nested sums corresponding to Yates' algorithm. For computing I'_h, J_h and J'_h, we need to compute additional similar products; these steps are accounted for in Algorithm 4.2, which represents the resulting haplotype frequency estimation EM-algorithm.

Finally, if we do not want to consider all the possible haplotypes, we set the corresponding haplotype frequencies in f to 0 when initializing the routine. This effectively eliminates the haplotypes from the summation without complicating the computations further.

To summarize, unlike in the $O(k^2)$ algorithm, it is not necessary to precompute the error factors $\varepsilon(G(H_i, H_j), \hat{G})$ for all genotypes present in the data and pairs of haplotypes H_i, H_j of length m, which would take $O(nm4^m)$ time, where n is the number of trios and m is the number of considered SNPs. If we denote the number of iterations until convergence by r, then the total time complexity of the fast algorithm becomes $O(rnk \log k)$.

4.3.3 Unrelated individuals

The case of unrelated data sets is simpler to evaluate. As mentioned, for genotypes $\hat{G}_1, \hat{G}_2, \ldots, \hat{G}_n$ sampled from n unrelated subjects, the likelihood is obtained by assuming independence:

$$L(\hat{G}_1, \hat{G}_2, \ldots, \hat{G}_n) = L(\hat{G}_1)\, L(\hat{G}_2) \cdots L(\hat{G}_n)\,.$$

Zou and Zhao [147] showed how to use an error model in the EM-algorithm for unrelated genotypes. The derivation of the algorithm is similar to that for trios, so we note only that following our notation, the M-step of the EM-algorithm becomes

$$f_h^{(1)} = \frac{1}{2n} \sum_{u=1}^{n} \sum_{i=1}^{k} \frac{f_i f_h (\varepsilon(G(H_i, H_h), \hat{G}_u) + \varepsilon(G(H_h, H_i), \hat{G}_u)}{\sum_{j=1}^{k} \sum_{j'=1}^{k} f_j f_{j'} \varepsilon(G(H_j, H_{j'}), \hat{G}_u)}.$$

This can easily be computed in time $O(nk^2)$, as long as we have precomputed the error factors for each pair of haplotypes. However, as the product in the numerator can be written out as a sum

Algorithm 4.2 *DD_EM* -subroutine for trios

Input: Trio data set $\mathcal{D}$, k potential non-deletion haplotypes, number of EM restarts r. Boolean value *deletion_case* for permitting deletion haplotypes,

Output: Estimated haplotype frequencies f and the data likelihood with the frequencies, $L(\mathcal{D})$

for *loop*$= 1, \ldots, r$ **do**
  $f \leftarrow$ random initialization s.t. $\sum f = 1$ and $f_i > 0$, $i = 0, \ldots, k$
  if not *deletion_case* **then**
    $f_0 \leftarrow 0$
    $s \leftarrow \sum_{h=1}^{k} f_h$
    $f_h \leftarrow f_h / s$ for all h
  end if
  while data likelihood has not converged **do**
    for t=1,. . . , n **do**
      $\tilde{\beta}_h \leftarrow \sum_{j'=0}^{k} f_{j'} \beta_{hj'}$ for all h {Yates' algorithm}
      $\tilde{\alpha}_h \leftarrow \sum_{i'=0}^{k} f_{i'} \alpha_{hi'}$ for all h {Yates' algorithm}
      $\tilde{\gamma}_h \leftarrow \sum_{j=0}^{k} f_j \gamma_{hj} \tilde{\beta}_j$ for all h {Yates' algorithm}
      $\tilde{\gamma}_h^J \leftarrow \sum_{j=0}^{k} f_j \gamma_{hj} \tilde{\alpha}_j$ for all h {Yates' algorithm}
      $I'_h(T_t) \leftarrow f_h(\sum_{i=0}^{k} f_i \alpha_{ih} \tilde{\gamma}_i)$ for all h {Yates' algorithm}
      $J'_h(T_t) \leftarrow f_h(\sum_{j=0}^{k} f_j \beta_{jh} \tilde{\gamma}_j^J)$ for all h {Yates' algorithm}
      $I_h(T_t) \leftarrow f_h \tilde{\alpha}_h \tilde{\gamma}_h$ for all h
      $J_h(T_t) \leftarrow f_h \tilde{\beta}_h \tilde{\gamma}_h^J$ for all h
      $L(T_t) \leftarrow \sum_{i=0}^{k} I_i(T_t)$
    end for
    $L^{loop}(\mathcal{D}) \leftarrow \prod_{t=1}^{n} L(T_t)$
    for h=0,. . . ,k **do**
      $f_h \leftarrow \sum_{t=1}^{n} (I_h(T_t) + I'_h(T_t) + J_h(T_t) + J'_h(T_t)) / L(T_t)$
    end for
    $f \leftarrow f / \sum_{h=0}^{k} f_h$
  end while
end for
Return the f associated with the loop maximizing data likelihood $L^{loop}(\mathcal{D})$.

of two products, both of these products can be computed by using essentially the same Yates' algorithm as we described for the case of trio data sets. The appropriate subroutine is described as Algorithm 4.3.

Hence, the time complexity of one iteration can be written as $O(nk \log k)$, where n is the number of unrelated individuals and not trios unlike in the previous section.

Algorithm 4.3 *DD_EM* -subroutine for unrelated individuals

Input: Data set $\mathcal{D}$ of unrelated individuals, k potential non-deletion haplotypes, Boolean value *deletion_case* for permitting the deletion haplotype and the number of EM restarts r.

Output: Estimated haplotype frequencies f and the data likelihood with the frequencies, $L(\mathcal{D})$

1: **for** $loop = 1, \ldots, r$ **do**
2: $f \leftarrow$ random initialization s.t. $\sum f = 1$ and $f_i > 0$, $i = 0, \ldots, k$
3: **if** not *deletion_case* **then**
4: $f_0 \leftarrow 0$
5: $s \leftarrow \sum_{i=0}^{k} f_i$
6: $f_h \leftarrow f_h / s$ for all h
7: **end if**
8: **while** likelihood not converged **do**
9: **for** $u = 1, \ldots, n$ **do**
10: Compute $v_h^u \leftarrow f_h \sum_{i=0}^{k} f_i \varepsilon(G(H_i, H_h), \hat{G}_u)$ for all h {Yates' algorithm}
11: Compute $v_0^u \leftarrow f_0 \sum_{i=0}^{k} f_i \varepsilon(G(H_i, H_0), \hat{G}_u)$.
12: $L(G_u) \leftarrow \sum_{i=0}^{k} v_i^u$
13: **end for**
14: **for** $h = 0, \ldots, k$ **do**
15: $f_h \leftarrow \sum_{u=1}^{n} \sum_{i=0}^{k} v_i^u / L(G_u)$
16: **end for**
17: $f \leftarrow f / \sum_{h=0}^{k} f_h$
18: **end while**
19: **end for**
20: Return the f associated with the loop maximizing data likelihood $L^{loop}(\mathcal{D})$.

4.4 Estimating the error probabilities

The miscall and null call rate parameters, τ and δ, which affect the error probabilities as described in Section 4.2.2 and Table 4.1, need be estimated based on the data given. The maximum likelihood estimate for the null genotype call rate δ is straightforward to estimate as the proportion of null genotypes over all genotypes.

The case of τ is more complicated. To this end, two different methods were tried. In the simpler case, we assumed that the parent genotypes were correct and only the child haplotypes could have been wrong. The focus was on the observation probability for informative trios where both parents were homozygotes, possibly with respect to different alleles, in which case the child genotype would be completely determined by the parent genotypes. Each SNP in each trio was considered separately.

Let us call $\mathcal{T}$ the collection of all triplets (F_t^s, M_t^s, C_t^s) in the data set that have homozygous parents and no null genotypes, and let $\mathcal{C}$, $\mathcal{F}$ and $\mathcal{M}$ be the children, fathers and mothers in these trios. In such a case and assuming no genotyping errors or Mendelian errors in the parents, the child genotype can be unambigously inferred. Let this inferred genotype be $C_t^{\text{inf.}}$. The children's genotypes' log-probability given their parents can thus be written as

$$\begin{aligned}
\log L(\mathcal{C}|\mathcal{M},\mathcal{F}) &= \sum_{t\in\mathcal{T}} \log \varepsilon(C_t^{\text{inf.}}, \hat{C}_t) \\
&= c_1 \log(1-\tau)^2 + c_2 \log(\tau^2) + c_3 \log(2(1-\tau)\tau) \\
&\quad + c_4 \log(\tau(1-\tau)) + c_5 \log((1-\tau)^2 + \tau^2) \\
&\quad + c_6 \log(1-\tau) + c_7 \log \tau,
\end{aligned}$$

where c_h are the number of trios for which the difference between the inferred child genotype $C_t^{\text{inf.}}$ and C_t results in the associated factor. The scenarios that result in each factor are listed in Table 4.1 but here we consider only cases where the observed genotype is not null, hence δ and $1-\delta$ are omitted from the factors. Note that when dealing with data that assumes there are no deleted SNPs in the parents either, then $c_6 = c_7 = 0$.

To find the value of τ to maximize this, we differentiate the likelihood $\log L(\mathcal{C}|\mathcal{M},\mathcal{F})$ for τ. The solutions to $\frac{\partial}{\partial\tau} \log L(\mathcal{C}|\mathcal{M},\mathcal{F}) = 0$

are the solutions of

$$\begin{aligned} 0 \;=\; & (-4c_1 - 4c_2 - 4c_3 - 4c_4 - 4c_5 - 2c_6 - 2c_7)\tau^3 + \\ & (4c_1 + 8c_2 + 6c_3 + 6c_4 + 2c_5 + 2c_6 + 4c_7)\tau^2 + \\ & (-2c_1 - 6c_2 - 4c_3 - 4c_4 - 2c_5 - c_6 - 3c_7)\tau + \\ & (2c_2 + c_3 + c_4 + c_7); \end{aligned} \tag{4.4}$$

for the solution to be feasible, we require that $\tau \in [0, 1]$. The true minimizing solution is found by testing all feasible solutions to the equation above and also the interval end-points 0 and 1.

An alternate method that was tried resembled the above procedure greatly. The SNPs in the data were considered to be independent and their allele frequencies estimated by the means of the same EM-algorithm depicted earlier in Algorithms 4.2 and 4.3. At the same time, the EM-algorithm was used to improve the estimate of τ. Finding $\tau^{(1)}$ to maximize the conditional expected log-likelihood $Q(f^{(1)}|f)$ described in Section 4.3.2 is otherwise the same as solving Eq. (4.4), but the definition of c_h has changed to be the sums of $\Psi_t(i, i', j, j')$ as they involve the different types of trios. More formally, let us define $U^s(\hat{C}_t^s, H_i^s, H_j^s)$ as vectors of zeros, except for one element that equals 1. The index of this element corresponds to the error table factor of $\varepsilon(G(H_i^s, H_j^s), \hat{C}_t^s)$ in c. For example, if the error is of a kind that has probability $2(1-\tau)\tau$, then it corresponds to factor c_3 and the index is set to 3. We now can write the joint update vector pertaining to child genotypes as

$$U_C = \sum_{s=1}^{m} \sum_{t=1}^{n} \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{j'=1}^{2} \sum_{i'=1}^{2} U^s(\hat{C}_t^s, H_i^s, H_j^s) \Psi_t^s(i, i', j, j'),$$

where $\Psi_t^s(i, i', j, j')$ equals $\Psi_t(i, i', j, j')$ computed for the one-SNP window of the SNP s. The vector U_C effectively contains expected counts of how many times each error type corresponding to a factor c_h occurs in the observations of child genotypes, under the assumption of independent SNPs. Note that in the preceding notation we did not allow for deletion alleles to be present due to the assumption of the SNPs being independent. The assumption that adjacent SNPs are independent decreases the accuracy of the deletion haplotype frequency estimate because the deletion alleles are no longer tied together to the same individuals and haplotypes. Deletions are

also expected to represent only a small part of the whole genome. This makes their inclusion in the estimate unnecessary.

Sums U_M and U_F representing expected error counts in mother and father genotype observations are defined analogously. Finally, maximising τ reduces to finding the solution to Eq. (4.4) with $c = U_C + U_F + U_M$, where the individual multipliers c_h are given by the elements of c. We note that each iteration of the EM-algorithm now takes time $O(nm)$ with the estimation of SNP allele frequencies included. Note that 2^m does not appear as a factor, because the SNPs are considered independent.

Computationally, the latter method is naturally the slower one, but it is also more accurate. To test the methods, COSI [104] and the best-fit parameters provided in the article [104] were used to generate 100 data sets in the European subpopulation without deletions with 250 SNPs and 100 trios. Simulated errors were then added to the data sets by using error parameters $\tau = 0.001$ and $\delta = 0.01$ to compare the accuracy of the two methods. As the result, the mean for EM-estimate of τ was 0.0010 with standard deviation of 0.000152, whereas the simpler estimate had mean of 0.0014 with standard deviation of 0.000248.

Note that in the case of data sets with deletions, the estimate becomes biased towards higher estimates of τ. This is because hemizygous deletions are more likely to introduce Mendelian inconsistencies into the data. These are perceived as miscalls, which results in the bias.

4.5 Estimating the significance

There are several methods of deciding from the results of the EM-algorithm whether the data supports the presence of a deletion or not. We take a closer look at two methods: likelihood ratio tests (e.g., [140, Ch. 13]) and k-fold crossvalidation (e.g., [48, pp. 214–217]). As third option, we consider using a data set screened to be (mostly) without deletion signal.

The exclusion and inclusion of the deletion haplotype in the haplotype frequency estimation produces two different models. The difference in parametrization is the addition of one variable to the former, this being the relative frequency of the deletion haplotype.

As the result of the EM-algorithm for estimating the frequencies, we gain a local maximum of the data likelihood. These maximized likelihoods can be used in the standard likelihood ratio test, explained for instance by Wilks [140, Ch. 13] and Ewens and Grant [32, Ch. 9.4]. This method was used also by Corona et al. [22] to give p-values for putative deletions. Kohler and Cutler [68] also use the likelihood ratio test to estimate the significance of deletions both on a per-SNP basis and then for the whole candidate deletion.

Let us consider two *nested models* M_0 and M_1 so that the parameter set of M_0, Θ_0, is a subset of that of M_1, Θ_1. These models correspond to the null hypothesis H_0 and the alternative hypothesis H_1, respectively. Let the maximum likelihood parameter vectors for these models then be θ_0 and θ_1 for the respective models. In the asymptotic case when sample size, in our case n, approaches infinity,

$$-2\log\left[\frac{L(\mathcal{D}|\theta_0)}{L(\mathcal{D}|\theta_1)}\right] \sim \chi^2(|\Theta_1 \setminus \Theta_0|)$$

as the size of the data, $|\mathcal{D}|$, approaches ∞. This can then be used to test whether to reject H_0. In the case of deletions and haplotype frequencies, Θ_0 is the set of random variables depicting the $k-1$ haplotype frequencies (the kth variable is completely determined by the other variables) and Θ_1 adds to that set the deletion haplotype frequency, thus having k elements. Hence, the test uses χ^2 distribution with one degree of freedom and tests whether the null hypothesis of no deletion being present should be rejected.

This test makes a number of assumptions as mentioned by e.g. Ewens and Grant [32, Section 9.4]: the parameters must be real numbers on some interval, and the maximum likelihood estimate is not a boundary point in the parameter space. However, with these assumptions we can compute the associated p-value. Small p-values suggest the presence of a deletion.

The algorithm for using the likelihood ratio test statistic is given as Algorithm 4.4. Note that both types of data, unrelated individuals and trios, can be used by this same algorithm separately, if the called algorithm *DD_EM* is substituted with either Algorithm 4.2 or 4.3.

To investigate the accuracy of this testing method, SNP data without deletions were generated with COSI [104]. From these synthetic data, empirical distributions of p-values under the null

Algorithm 4.4 Deletion detection algorithm utilizing the likelihood ratio test statistic.

Input: Genotype data set $\mathcal{D}$.

Output: The estimated deletion frequency f_0^1 and the likelihood ratio test statistic t.

1: $[f^0, L^0] \leftarrow DD\_EM(\mathcal{D}, false)$
2: $[f^1, L^1] \leftarrow DD\_EM(\mathcal{D}, true)$ $\{f_0^1$ is the estimated deletion frequency$\}$
3: Return f_0^1 and the likelihood ratio test statistic $t = -2 \log \frac{L^0}{L^1}$.

hypothesis were formed with the number of trios ranging from 30 to 500, and the window size, m, from 2 to 8.

50 data sets of 500 kb in length the under the "European" population model were generated. To simulate SNP ascertainment in the synthetic data sets, two different schemes were used: one was to use an adaptation of tag-SNP selection algorithm of Carlson et al [17], and the other was a simulated panel of 48 haplotypes. The adaptation of the algorithm by Carlson et al. is depicted in Algorithm 4.5. The main difference is that the mean SNP spacing and the threshold for minimum r^2 were fixed, and that in dividing the SNPs into bins of high LD, only one SNP was added to the tag-SNP collection.

Algorithm 4.5 An adaptation of the SNP tagging algorithm of Carlson et al. [17]

Input: A haplotype data set.

Output: A set of SNPs selected for genotyping.

1: Remove SNPs with MAF below 0.05.
2: Select SNPs so that their mean distance is 2 kb.
3: Sample 120 haplotypes at random and ignore the rest.
4: Again remove SNPs with MAF below 0.05 in the now smaller data set; $S \leftarrow$ the remaining SNP set.
5: **while** S not empty **do**
6: $\quad k = \text{argmax}_i \sum_j r^2(i, j)$
7: $\quad$ Add SNP k to genotyped SNPs.
8: $\quad$ Eliminate SNPs i for which $r^2(k, i) \geq 0.7$ from S.
9: **end while**

To compare the effect the SNP selection schemes have on the

score distributions, in addition to the method of Carlson et al. [17], selection of SNPs with a panel that was included in the final data set was also simulated. 48 haplotypes in the parental haplotypes were selected at random, and all SNPs with two alleles present in the panel haplotypes were included in the data set after removing those with MAF below 0.05 and removing enough SNPs to make the mean SNP distance 2 kb. Note that the tag-SNP algorithm and the panel ascertainment method produced data with different mean SNP distance, which also means that windows spanned different lengths in the data. The miscall and no call errors were modelled using the error model in Section 4.2.2 with $\tau = 0.001$ and $\delta = 0.01$.

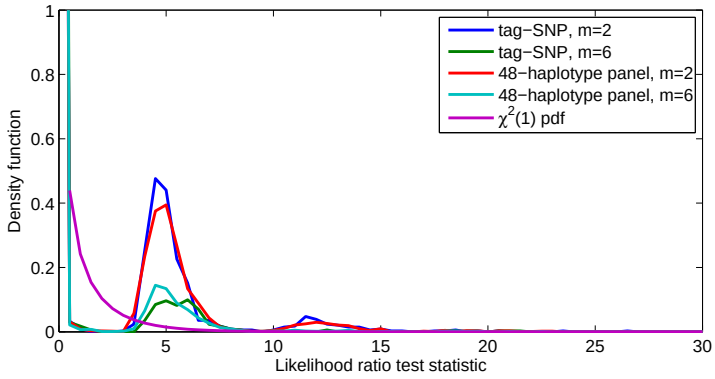
The test statistic values were computed from sliding windows of fixed width, so one data resulted in multiple observations of the test statistic. The number of such windows in each data set was dependent on the window size m. As seen in Figure 4.2, the empirical distribution of $-2 \log \frac{L(H_0)}{L(H_1)}$ does not strictly follow the χ^2-distribution with one degree of freedom. The difference between the theoretical and empirical probability density functions possibly decreases with increasing the window size, which also corresponds to growing region covered by the window (Figure 4.2 (a,c)). It is understandable that with the window size increasing it becomes less likely for random genotyping errors or pure chance to result in a false discovery of a deletion.

The number of trios also has a clear impact on the accuracy of the approximation (Figure 4.2 (b)). The used SNP screening method has a notable effect on the test statistic distribution as well.

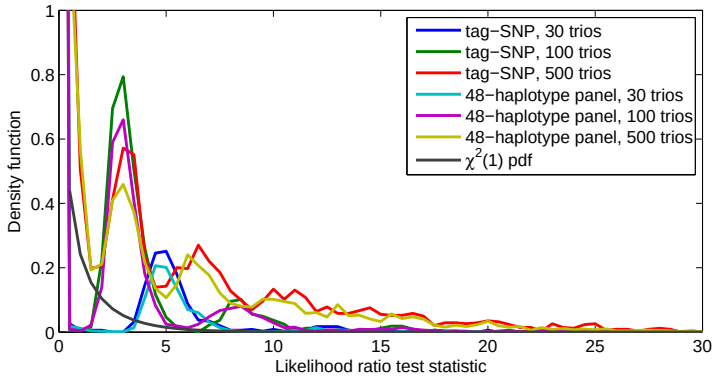
Regardless, we have no reason to expect that high-scoring values of the test statistic are less significant than lower-scoring values, assuming the same sample and window size. We can still select test statistic thresholds for accepting the alternate hypothesis; we only are not certain of the false positive rate with real data for some fixed threshold.

Another frequently used tool for model selection is k-fold cross validation. In the former, the data is partitioned in k equally large subsets; let us denote these sets now $\mathcal{D}_i$ and call them *test data*.

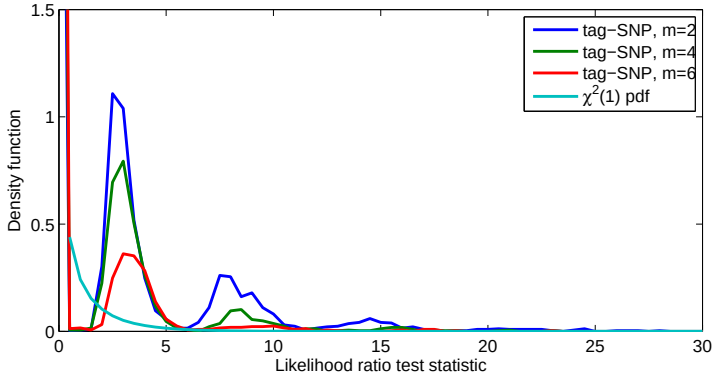
Let us now use the EM-algorithm for each *training data* set $\mathcal{D} \backslash \mathcal{D}_i$ to estimate the haplotype frequencies Θ_1^i and Θ_0^i for with and without deletion haplotype, respectively. Given these two haplotype fre-



(a) Varying window size and ascertainment method with 30 trios



(b) Varying trio count, $m = 4$



(c) Varying m, 100 trios with tag-SNP algorithm

Figure 4.2: Empirical likelihood ratio test score distribution, computed from 50 replicated SNP data generated by COSI without deletions, plotted against the $\chi^2(1)$ probability density function. The total number of SNPs in each replication was 250.

quency sets, we then compute the likelihood of $\mathcal{D}_i$. The test value these likelihoods give us is

$$t_{\mathrm{CV}} = \frac{1}{k}\sum_{i=1}^{k}(\log L(\mathcal{D}_i|\Theta_1^i) - \log L(\mathcal{D}_i|\Theta_0^i)).$$

The resulting algorithm is given as Algorithm 4.6.

Algorithm 4.6 Basic k-fold crossvalidation framework for DelDec.

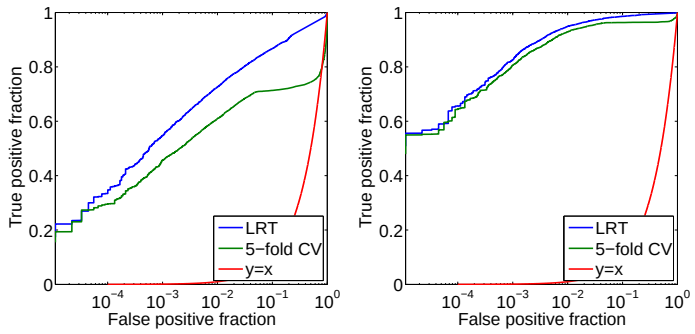
Input: Genotype data set $\mathcal{D}$ and the number of folds in crossvalidation, k.

Output: The estimated deletion frequency f_0^1 and the test statistic.

1: Partition $\mathcal{D}$ randomly in k parts $\mathcal{D}_1, \ldots, \mathcal{D}_k$.
2: $t_{\mathrm{CV}} \leftarrow 0$
3: **for** $i = 1, \ldots, k$ **do**
4: $\quad [f^{0,i}, L^i(0)] \leftarrow DD\_EM(\mathcal{D} \setminus \mathcal{D}_i, false)$
5: $\quad [f^{1,i}, L^i(1)] \leftarrow DD\_EM(\mathcal{D} \setminus \mathcal{D}_i, true)$
6: $\quad t_{\mathrm{CV}} \leftarrow t_{\mathrm{CV}} + \log L(\mathcal{D}_i|f^{1,i}) - \log L(\mathcal{D}_i|f^{0,i})$
7: $\quad f_0^1 \leftarrow f_0^1 + \frac{1}{k} f_0^{1,i}$
8: **end for**
9: Return the test statistic $\frac{1}{k}t_{\mathrm{CV}}$ and deletion frequency f_0^1.

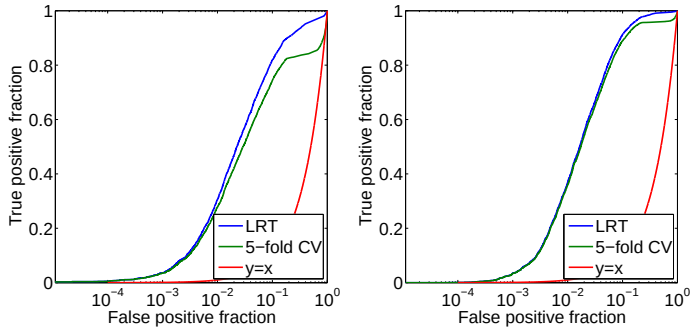
In practice, 5-fold crossvalidation performs typically as well or worse than the likelihood ratio test. This was tested by using the synthetic data sets described in Section 5.3.1. Figures 4.3 and 4.4 display some example ROC curves generated under various conditions. If the deletion is sufficiently frequent ($f_0 \geq 0.1$) and window size at least 4, there is no major difference in the performance between the two methods. With smaller window size there is a difference (Figure 4.4). There is also a difference if the amount of data n is small or if the true proportion of deletions is small. The ROC curves are drawn on per-SNP-detection accuracy by using the mean method (described in Section 4.6).

Finally, the third option for estimating significance stems from the presence of real-world SNP data. Recall that the presence of deletions increases the number of no call genotypes and the distance from Hardy–Weinberg equilibrium. Both of these factors have been used as quality control (QC) criteria for eliminating poorly genotyped SNPs from data sets [128, 129]. Because such QC can remove



(a) $n = 30, f_0 = 0.05$, window size 4

(b) $n = 30, f_0 = 0.1$, window size 4



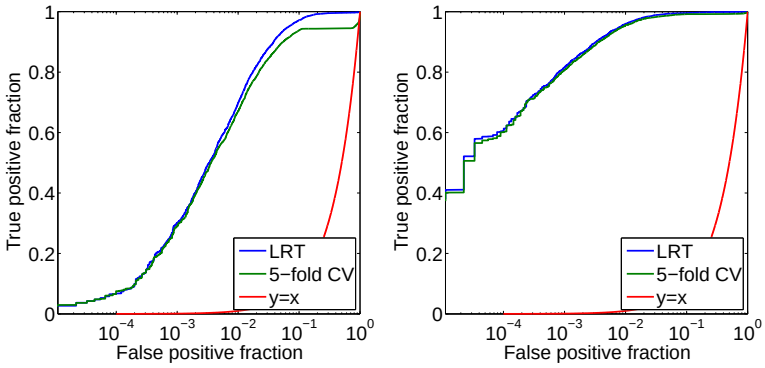
(c) $n = 500, f_0 = 0.005$, window size 4

(d) $n = 1000, f_0 = 0.005$, window size 4

Figure 4.3: ROC-curve comparisons of 5-fold CV and likelihood ratio tests (LRT) under varying synthetic data parameters.

information about deletions, it is sensible to perform the deletion detection scheme described in this chapter on data sets that have not undergone such a QC process.

The QC process, however, can be also be used to improve a deletion significance estimation scheme: the filtered data set can be expected to have weaker deletion signals, but the QC process most likely does not remove all of it. Hence the score distribution in this data set is a combination of both deletion and deletion-free signals, but is closer to deletion-free signals than the unscreened data set. This distribution can then be used as a conservative estimate for translating the p-values the likelihood ratio test produces into p-value estimates that might be closer to their real values than the



(a) $n = 100$, $f_0 = 0.05$, window size $m = 4$

(b) $n = 100$, $f_0 = 0.05$, window size $m = 4$

Figure 4.4: ROC-curve comparisons of 5-fold CV and likelihood ratio tests (LRT) under different window sizes.

χ^2 approximation.

The overall idea is to use a data set with significantly weaker signals for the presence of deletion to form a closer approximation of the underlying distribution of the test statistic. Note that this approach is feasible only for very large data sets, such as the HapMap data, as the score histogram need be defined at a sufficiently fine scale.

4.6 Determining deletion end-points

Merely reporting the presence of a deletion is rarely enough, as the approximate location of the SNP should also be given.

The haplotype frequency estimation method presented in Sections 4.2 and 4.3 does not directly tell where the deletion breakpoints are, as also windows partially covering regions flanking the deletions can have high test statistic values. In the method of Kohler and Cutler [68] this is not a problem, as their method defines the putative deletion starting from the evidence from single SNPs, expanding the putative deletions from each SNP and then combining the expanded deletion frames.

By comparison, Corona et al. [22] join overlapping windows with

the likelihood test ratio score higher than a pre-determined value, and report the positions of the first and the last SNPs within the window. As our method is also based on the method of Corona et al., it also uses statistics for windows rather than single SNPs. Therefore we want to address the problem for a deletion status prediction for each SNP, as in our approach the windows need to be joined together to produce deletion candidates longer than at most 10 SNPs.

Other methods for the same purpose were also considered. The simplest case was the estimation based on one-SNP windows, which can be summarized as computing the per-SNP deviation from the Hardy–Weinberg equilibrium. The shortcoming of this approach is that it lacks power in the case of SNPs with very low MAF, as it completely ignores the possible presence of deletions on either or both sides of the SNP. This option was not tried out in the experiment.

Alternatively, we approach the problem by taking the solution of Corona et al. as a starting point. We examine whether there are SNPs that are inside a deletion polymorphism within the window. For Corona et al., the window size varied from 1 to 40 SNPs. In our approach, we use only a fixed-size sliding window over the genome to produce a scan of the dataset, and included all potential haplotypes in the consideration, i.e., $k = 2^m$.

Three different end-point determination methods were considered. The simplest option was to label a SNP deleted if it was contained in at least one window with a likelihood ratio test score over a given threshold. We call this the single-hit method; it was also essentially the same as Corona et al. [22] used in their work to combine windows. The second option required at least half of the windows containing the SNP to score above the given threshold; this is called the majority-vote method. Finally, the third option considered was to investigate the arithmetic mean of the scores of the windows containing the SNP and to label the SNP as deleted if the mean score was over a threshold. In the following, we call this the mean method.

The three methods were tested on synthetic data sets, the same ones as used in Section 5.3.1 to evaluate the performance of our deletion detection method. The mean SNP spacing was 2 kb and the deletion length was 20 kb in a simulated segment of 250 kb. The

used criteria were the fractions of false positives and true positives when predicting the SNP-wise deletion status.

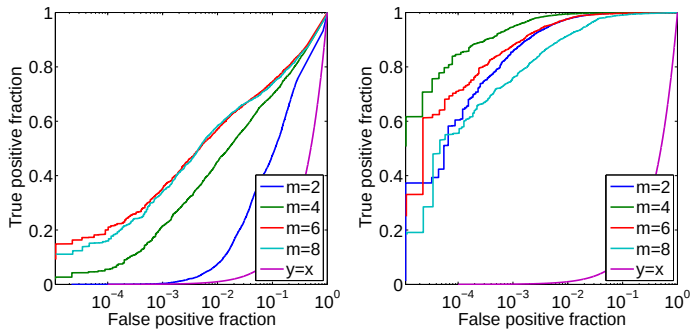
The results are shown in Figures 4.6 and 4.5. Long windows ($m \geq 6$) worked well in these synthetic experiments only in cases where there were a large number of trios ($n \geq 100$) and the deletion was rare ($f_0 \leq 0.01$). The good performance in these cases might be due to the decreased variance in the test score due to the score being more robust against random noise. When the deletion is rare, the signal is lost in the random noise when using small windows. In contrast, in longer windows the consistent deletion patterns increase the likelihood ratio test score. Therefore, in the case of rare deletions, the long windows have a better chance of detecting the presence of a deletion and the short windows cannot compete despite their more accurate deletion end-point detection. In shorter windows, the deletion signal from one end of the window does not increase the score of the SNPs in the other end.

Of the three different methods, the mean method seems to perform well in data sets of moderate size (Figure 4.6(a) and (b)). By selecting this method we get Algorithm 4.7, which we call Deldec-Scan, for scanning over whole-genome data sets.

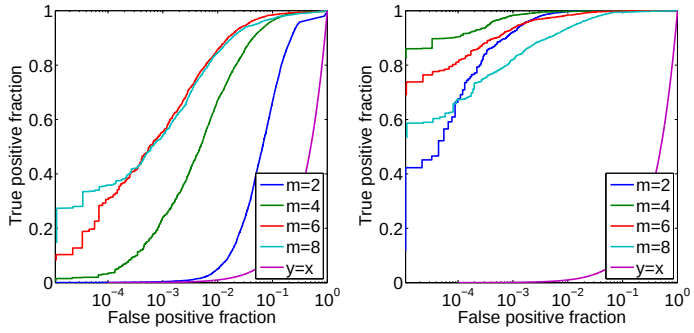
Algorithm 4.7 Deldec-Scan -algorithm for detecting deletions in whole-chromosome SNP data sets. LRT($\cdot$) corresponds to Algorithm 4.4 with 2^m different potential haplotypes.

Input: Window size m, genotype data $\mathcal{D}$
Output: Deletion candidate regions

1: **for** $i = 1$ to $n - m + 1$ **do**
2: $s_i \leftarrow$ LRT(genotypes of SNPs in i through $i + m - 1$ in $\mathcal{D}$).
3: **end for**
4: **for** $i = 1$ to n **do**
5: $c \leftarrow$ mean of SNP deletion scores s_{i-m+1} through s_i (excluding indices below 1, that is, SNP positions outside the data).
6: **if** $c >$ threshold **then**
7: Mark SNP i as deleted.
8: **end if**
9: **end for**
10: Join adjacent SNPs marked as deleted together as deletion candidates.



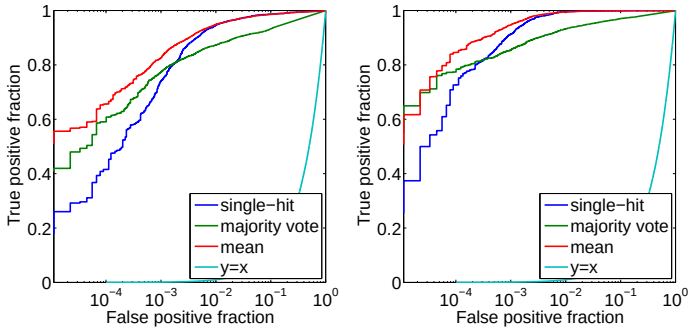
(a) $n = 100$, $f_0 = 0.01$ (b) $n = 100$, $f_0 = 0.1$



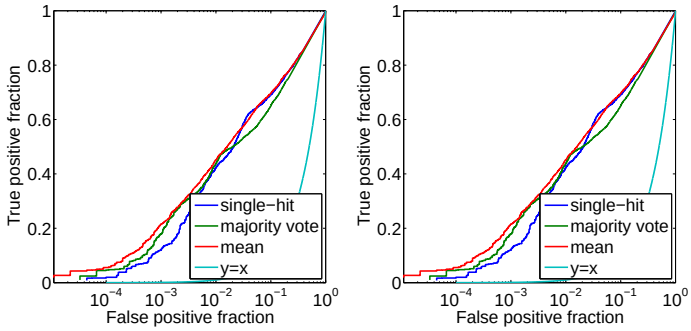
(c) $n = 500$, $f_0 = 0.01$ (d) $n = 500$, $f_0 = 0.1$

Figure 4.5: ROC-curve comparison of accuracy of deletion end-point estimation in terms of correctly labelled SNPs for different window sizes. The used SNP-wise deletion status prediction method was the mean method. Here n is the number of trios, f_0 is the deletion haplotype frequency and m is the window size.

Because the described methods only detect the presence of a deletion spanning certain set of SNPs rather than detecting which specific SNPs have a deletion allele, the main purpose for reasonably good end-point location estimates is their use in designing experimental validation tests by, e.g., fluorescent *in situ* hybridization, quantitative PCR or PCR amplification as done in [81]. In such case, the segment selected for sequencing spans the estimated deletion segment and the flanking regions. In this context, the determination of exact deletion end-points loses some of its significance when the SNP density near the estimated deletion end-points



(a) $n = 30, f_0 = 0.1, m = 4$ (b) $n = 100, f_0 = 0.1, m = 4$



(c) $n = 100, f_0 = 0.01, m = 4$ (d) $n = 500, f_0 = 0.01, m = 4$

Figure 4.6: ROC-curve comparison of accuracy of deletion end-point estimation in terms of correctly labelled SNPs for different methods. Here n is the number of trios, f_0 is the deletion haplotype frequency and m is the window size.

is high. In such case, it is less important to be completely accurate of the deletion end-point, as slight inaccuracy will not likely cost much more in the experimental validation process.

4.7 Discussion

A method for detecting the presence of deletions from SNP genotype data has been presented. Although each iteration of the EM-algorithm takes $O(k \log k)$ time in terms of different haplotypes k for dense haplotype sets, i.e. $k = O(2^m)$, windows of reasonable length

($m \leq 10$) can be computed on modern computers sufficiently fast for high-throughput processing. Such an upper limit on the window length is not a significant limitation for whole-genome scans, as with a simple routine that uses the algorithms 4.2 and 4.3 as subroutines we can scan over several million SNPs with reasonable computational resources. We expound these details of applicability, i.e., computation time and accuracy, in Chapter 5.

Instead of using previously defined error models [22, 68], we presented a third one in Section 4.2.2. It can be argued that the high number of parameters utilized by Kohler and Cutler's [68] model might make their model more suspectible to overfitting, but to what extent is not known. The potential overfitting in [68] is likely not an issue, as new high-throughput methods produce large amounts of SNP data which can be used to estimate the parameters accurately. As was shown in the experiments in [68], already rather few SNPs in the data set were sufficient for estimating the parameters so that the deletion predictions were accurate. Which error model works best is a logical question for future work and discussion, but will not be further addressed in this thesis.

Let us now briefly review the main differences between the algorithm of Corona et al. [22] (labelled 'CRE-method' in this section) and Deldec-Scan. Both algorithms use the same underlying data model of haplotypes. However, CRE-method first estimates initial haplotype frequencies by a haplotyping program and then scales the haplotype probabilities to incorporate also the deletion haplotype. This permits them to use longer haplotypes ($1 \leq m \leq 40$) than Deldec-Scan can ($m \leq 10$). This scaling approach was devised originally to avoid the high time complexity of the EM-algorithm, as were additional requirements for the investigated regions, i.e., a limited number of haplotypes within the window and at least one Mendelian error consistent with a deletion. We solved the same problem by using Yates' algorithm and constraining ourselves to a small window size m.

CRE-method then combines the windows that have a p-value below a threshold to form a nonoverlapping list of predictions. This is essentially the same as the single-hit method for deciding SNP deletion status. Section 4.6 lists two other ways to decide on SNP deletion status. Of these, the mean method is experimentally shown to perform better than single-hit method and as the method of

choice for Deldec-Scan.

While detecting the general presence of deletion is relatively straightforward, determining the end-points of the deletion accurately represents a problem. To this end, we compared three different methods to use in a sliding window approach. All these methods are rather ad-hoc in nature, which is not a desirable trait.

The question of accurate end-point detection might not be important enough in practice to warrant much more attention. Assuming the SNP data sets are dense, misestimating the deletion end-point by one or two SNPs does not result in a considerably larger number of genotyped basepairs in experimental verification of the candidate deletions.

CHAPTER 5

Experiments

The methods developed in preceding chapters were tested on synthetic data sets to evaluate their statistical power and specificity. The methods were also used to create a list of putative inversions and deletions from HapMap [128, 129] data sets.

5.1 InvCoal as an inversion simulator

To investigate whether InvCoal is more accurate in simulating inversions than ms [56], we experimented on the simulators for five known inversion polymorphisms. The idea was to use a statistic sensitive to the presence of an inversion to measure how well the simulators can reproduce the signal the summary statistic attempts to detect.

To make ms more comparable to InvCoal, ms was parametrized to use a two-population model where one population has a constant size and the other underwent exponential population growth until the lineages in that population were moved to the main, ancestral, population. There was no migration between these populations.

InvCoal uses otherwise the same population models as ms with the exceptions of the inversion spanning only as many basepairs as the real inversion, recombinations being suppressed and the populations not being completely separate, especially outside the inversion. Note that ms cannot use the information on the inversion position, as the population subdivision can only span the whole simulated segment or not be present at all.

Normalized bicomponent score, NBS, is a score described in Section 3.2. It attempts to detect inversions by the haplotype subdivision into two distinct populations. Note that ms can produce segments with high NBS in the given model, as in practice the models used in InvCoal and ms within the inversion are very similar from the perspective of NBS.

In the Database of Genomic Variants [59] there are several different inversions listed for different chromosomes. The inversions for simulation studies were chosen from these by the following criteria for each of the three HapMap data sets (CEU, YRI and JPT+CHB) separately.

First, the inversion had to be between 250 kb and 1.5 Mb long. Shorter inversions would not show up on NBS and longer ones with their flanking regions are too long to simulate with InvCoal.

Now, let the inversion length be l_1 kb long and $l_2 = \min(l_1, 500)$. The second criterion was that the inversion region had to contain at least 20 SNPs with MAF higher than 0.05, and also the region $(500 + l_2)$ to 500 kb before the inversion had to contain at least 20 SNPs with MAF higher than 0.05. NBS requires a sufficient amount of SNPs for valid results. The latter requirement is because for the actual test statistic, we compute NBS also in that region.

Third, the inversion was not allowed to intersect with another known inversion. This is because InvCoal does not model regions with multiple inversion events.

Fourth, the NBS within the inversion had to be at least 0.5. This is a high value. One reason for this is that although all the inversions in the database are experimentally validated, their population frequencies are typically unknown. As such, high NBS score suggests that the inversion is common enough to be detected and of the type NBS can detect and InvCoal simulate. NBS cannot detect all inversions, possibly because inversions are not frequent enough or they are recurrent. InvCoal does not try to simulate recurrent inversions, so it is sensible to limit the focus to inversions NBS alone can detect. It is also possible that InvCoal lacks other features that pertain to simulating inversions in particular. We acknowledge that this means that the results of these experiments are not indicative of all inversions but only a specific subset of inversions.

In the end, there are 4 inversions meeting these criteria in the HapMap data set, one of which is in 2 populations, yielding a total

of 5 inversions. If we include also the inversions that meet all other criteria except the one of high NBS, the number of inversions increases to 37 when inversions in different populations are considered distinct.

NBS is designed to peak within inversions and stay low outside inversions. Therefore we choose as the statistic of interest

$$\Delta(\mathcal{D}) = \mathrm{NBS}(\mathcal{D}_{\mathrm{in}}) - \mathrm{NBS}(\mathcal{D}_{\mathrm{out}}),$$

where $\mathcal{D}_{\mathrm{in}}$ is the data for the SNPs within the inversion and $\mathcal{D}_{\mathrm{out}}$ is the data for the SNPs in the l_2-kb long region before the inversion. In the case of real data, $\mathcal{D}$ stands for $\mathcal{D}_{\mathrm{in}}$ and $\mathcal{D}_{\mathrm{out}}$ as a pair. Ideally, the former should be high and the latter low to reflect the inversion status of the segment. A simulator should produce this effect equally strongly as it is present in the real data. The better a simulator can reproduce this difference between the segments, the better the simulator can be considered in this aspect.

The simulators were used to simulate data sets of length $2 \times l_2 +$ 500 kb. In the case of ms, the first l_2 kb were used as the region outside the inversion and the last l_2 kb as the inversion. The same applied to InvCoal as well, but the inversion position was also given as a parameter.

The parameters for the simulators need to be chosen to fit each inversion and simulator separately. These parameters were the recombination rate r, ancestral effective population size N_e^{A}, inversion age, number of inversion haplotypes in the data set and the inversion population size. The value of the last parameter, f, is transformed into the inversion population effective population size as $N_e^{\mathrm{A}} f/(1-f)$. The parameters are listed in Table 5.1 with their possible values.

The parameter fitting was done by using a greedy search method by updating the parameters one at a time for several iterations. Once a parameter was chosen for updating, three data sets were simulated with each potential parameter value. The mean NBS difference of the three data sets was compared to the difference in the real inversion. The parameter by which the difference between these two was minimized was chosen as the updated parameter value. If a generated data set had less than 20 SNPs in either of the two relevant regions, the associated parameter value was not allowed to become the new value of the parameter.

Table 5.1: The parameters and their ranges investigated in the comparison of InvCoal and ms outputs to a known inversion. The size of the inversion population is N_e^{A} times the parameter inversion population size. The number of haplotypes in the data set is n.

Parameter name	Parameter values
r	$10^{-10}, 1\times10^{-9}, 2\times10^{-9}, \ldots, 3\times10^{-8}$
N_e^{A}	2,500; 3,000; ...; 15,000
Inversion age	5,000; 10,000; ...; 150,000
Inversion haplotypes	1, 2, ..., $n-1$
Inversion population size parameter f	0.01, 0.02, ..., 0.99

The parameters were updated in five rounds, each of which updated each parameter exactly once in a random order.

Once the best-fit parameters were found, 500 data sets were generated by both simulators. Let $\mathcal{D}^{*,i}$ be the ith simulator-produced data set for simulator $*$, which is either ic (InvCoal) or ms (ms). The mean of the differences between the NBS scores inside and outside the generated inversion

$$\bar{y}_* = \frac{1}{500}\sum_{i=1}^{500}\Delta(\mathcal{D}^{*,i})$$

was computed.

The actual test statistic is

$$\tilde{\lambda}(\mathcal{D}) = |\Delta(\mathcal{D}) - \bar{y}_{\mathrm{ms}}| - |\Delta(\mathcal{D}) - \bar{y}_{\mathrm{ic}}|. \tag{5.1}$$

Here, $\Delta(\mathcal{D})$ is the difference computed from the real data and not one of the simulations.

This is the difference between the mean errors produced by ms and InvCoal. The larger the statistic is, the better InvCoal performed in comparison to ms.

The difference alone does not reveal how significant the value of the statistic is. It is likely that the expected value of $\tilde{\lambda}$ is positive over data sets without inversions, if we assume that NBS in the two regions are independently and identically distributed. Note that the average difference in NBS produced by ms is 0, as the simulator models the two regions identically. Let us now assume

that $\Delta(\mathcal{D})$ is positive due to random variation. In this case, InvCoal can model this variation whereas ms cannot, resulting in a positive expectation for the statistic in this case. If $\Delta(\mathcal{D})$ is negative, then InvCoal should be able to find such parameters that the inversion population is very small and consists of few haplotypes, in which case the mean NBS difference should be close to 0.

For this reason, samples were generated from the null distribution of the statistic with the null hypothesis that the inversion was actually generated by ms by using the fitted parameters. First, ms was used to generate several data sets with the fitted parameters. The simulated data sets were $(2 \times l_2 + 500)$ kb in length, i.e., it was as long as the model segment in the real data sets. From each data set, the difference between the NBS scores computed from the first l_2 kb and the last l_2 kb in the synthetic data set were computed; let this be called y_i for the ith data set. The difference y_i was substituted as $\Delta(\mathcal{D})$ in Eq. (5.1) for the next step.

Because the statistic $\tilde{\lambda}$ uses fitted parameters to compute the mean of the statistics, the parameters for both InvCoal and ms have to be fitted again by using y_i as the substitute for the statistic computed from the real data. The next step was to produce 500 data sets with these newly fitted parameters. These data corresponded to the data sets $\mathcal{D}^{*,i}$, i.e., they were used to compute the mean differences produced by the simulators. By computing the difference in the accuracy of ms and InvCoal with respect to the substitute statistic, we gain a sampled point from $\tilde{\lambda}(\mathcal{D})$ under the null hypothesis of ms and InvCoal producing the same difference in NBS. These samples were then used to compute estimates for the p-value for the difference. The estimated p-values are listed in Table 5.2. The number of y_i investigated for each inversion are the denominators listed in the column $\hat{p}$.

In the case of all these 5 inversions, InvCoal was more accurate in simulating the data. The small value of $\tilde{\lambda}$ for the chromosome 11 inversion in the CEU data set is explained by the NBS being high also in the region outside the inversion.

The observed p-values are not by themselves sufficient to judge on whether InvCoal is better than ms, because the 5 inversions were carefully selected for further examination, which likely introduces bias.

For this reason, we consider the effect of the fourth filtering cri-

Table 5.2: Inversions by which ms and InvCoal were compared. The column $\hat{p}$ lists the proportion of null simulations where $\tilde{\lambda}$ was higher than the one observed for real data. The denominator is the number of points from which the p-value estimate was computed. The note $< 1/352$ means none of the simulations produced a $\tilde{\lambda}$-value higher than the one observed with real data.

Inversion	Pop.	$\tilde{\lambda}$	$\hat{p}$
chr4:171,552,938–171,850,814	CEU	0.3454	2/927
chr7:64,246,951–64,686,726	CEU	0.3472	2/426
chr11:50,047,247–50,337,552	CEU	0.0617	25/150
chr11:50,047,247–50,337,552	JPT+CHB	0.1731	6/158
chr17:40,899,921–41,989,253	CEU	0.3707	$< 1/352$

terion that required NBS to be high for the inversion to be investigated. This is done by considering us to have made 37 tests, but we know the actual p-values of only 5 of them and assume the p-values of the remaining 32 to be sufficiently large not to be considered significant.

The time consumption of producing sample points for the distribution of $\tilde{\lambda}$ for the listed inversions is high, for the fastest case (chr4:171,552,938–171,850,814) approximately one day per point on a system utilizing 7 CPU cores, each running at 2.53GHz. This was the reason why only a subset of selected inversions were investigated and also the reason why the number of null simulations per inversion was low.

Instead of showing that all the investigated inversions are simulated better by InvCoal than ms, the goal is to show that InvCoal is significantly more accurate in simulating at least some inversions. To bypass the effect of the filtering criterion in its entirety, we can try using the Bonferroni correction. To prove the claim, a p-value of below $0.05/37 \approx 0.0014$ is needed for the p-value to be below 0.05 due to the multiple testing correction. To achieve this, at least 740 points under the null hypothesis would need to

be computed to produce an estimate of the p-value with a significant value. The inversion with most computed samples from the null distribution, however, gives p-value estimate of $2/927$, which in turn corresponds to a Bonferroni-corrected p-value of 0.0798. The Bonferroni-corrected p-values for the other inversions are at least 0.105 due to the low number of computed samples.

An alternative approach to investigate these results is the false discovery rate (FDR) [10]. In brief, FDR is the expected proportion of false positives out of all positive predictions for a p-value threshold. If we fix the FDR limit at 0.125, three inversions with the smallest p-values are accepted with the procedure given by Benjamini and Hochberg [10] to control the FDR.

Storey [119] discusses a method of estimating the FDR for a fixed p-value threshold t. The equation in question is

$$\widehat{\mathrm{FDR}}(t) = \frac{\hat{\pi}_0 m t}{\#\{p_i \leq t\}}, \tag{5.2}$$

where m is the number of tests, p_i the associated p-values and $\hat{\pi}_0$ is the estimated proportion of the 37 tests for which the null hypothesis holds. Let us assume that all the inversions that were not investigated have p-value of 1. If we conservatively approximate $\hat{\pi}_0 = 1$ and fix $t = 0.01$, we have

$$\widehat{\mathrm{FDR}}(0.01) = \frac{37 \times 0.01}{3} \approx 0.123.$$

This means that the expected number of false positives out of the 3 significant results is 0.37, which is notably smaller than the number of predicted positives. Storey [119] also gives a formula for estimating for the pFDR (positive FDR) [119, 118], in which the expectation in FDR is conditioned on the event that at least one null hypothesis was rejected. The estimate for pFDR, gained from Eq. (5.2) by dividing it with $1 - (1 - t)^m$, is approximately 0.397. The FDR-estimate, however, is small enough to lend credence to the claim that some inversions of the 3 with the uncorrected p-value at most $1/100$ are better simulated by InvCoal than by ms in terms of $\tilde{\lambda}$. Both estimates have their points compared to the other. The estimate for pFDR is possibly too high due to the conservative estimate of $\hat{\pi}_0$, as Storey [119] reports the pFDR estimate tends towards $\hat{\pi}_0$, but on the other hand the number of tests is low which

gives a high probability for the case that no null hypothesis would be rejected. This makes the accuracy and use of $\widehat{\text{FDR}}$ questionable.

Let us then consider how likely it is that we have at least 3 false positives with the uncorrected significance level of 0.01. First, let us assume that the 37 tests are independent and each one, if investigated as described above, would result in InvCoal being labeled the better one with probability of 0.01, that is, by pure chance with p-value threshold of 0.01. The number of false positives with the fixed p-value threshold would then be binomially distributed with parameters 37 and 0.01 (e.g., [24]). In this model, the probability of having at least 3 false positives out of the 37 trials by summing the tail of the point mass function of the binomial distribution is 0.0060. According to this test, it is likely that InvCoal is better than ms in simulating at least some inversions. Even if the p-value threshold for single inversions is raised to 0.022, the p-value for the number of false positives would remain below 0.05. With single-inversion p-value threshold of 0.05, the p-value for the number of false positives is 0.1119.

Note that it is unknown how significant a presence the 37 inversions have in the HapMap data sets. It is possible that some populations do not have an inverted haplotype or have them in such quantities that they can be detected from SNP data, in which case keeping the number of tests at 37 results in conservative estimates.

Furthermore, this investigation utilized only one test statistic. It is possible that there are other statistics that may produce better results. These statistics could involve, for instance, Sindi and Raphael's [112] scoring for detecting inversions.

As a third point, not all real inversions appear to show the four-field pattern NBS tries to detect and InvCoal produces, possibly due to the assumption that inversion events are unique. As such, InvCoal should not be used as a simulator for all inversions there are but a specifically behaving subset of them. Further development of the simulator is hence a relevant task. Nonetheless, as a conclusion, the experiments show there are inversions that InvCoal can simulate better than ms.

5.2 Inversions

In our experiments here, the NBS-Scan algorithm, described in Section 3.5, was tested as a scoring for detecting inversions, first in synthetic data and second in real HapMap Phase III data sets.

5.2.1 Ascertainment and tag-SNP algorithms

Large whole-genome SNP data sets, such as the Perlegen data set [50] and HapMap data sets [128, 129], often use different and varying SNP ascertainment procedures. For instance, the ascertainment correction, or the process of removing the bias produced by the ascertainment process into the data, of the HapMap data set in particular is difficult because the scheme changed as the database was being built [19].

Genome-wide association studies typically use a set of SNPs chosen based on their LD or ability to help impute the alleles of other SNPs. It is interesting if the methods for detecting structural variants from SNP data can be used also on such data sets.

For this reason, the effects of different SNP selection schemes on the performance of the detection methods were investigated by using either:

- 48-haplotype panel ascertainment,
- the greedy tag-SNP selection algorithm adapted from Carlson et al. [17].

The panel ascertainment was simulated in a straightforward manner. The panel is made of a randomly selected subset of 48 haplotypes of all the haplotypes in the data. If a SNP is biallelic in the subset, the SNP is included in the data set.

The latter algorithm for tag-SNP selection was previously described as Algorithm 4.5 in Section 4.5.

5.2.2 Generating synthetic data

The overall goal of the synthetic data simulation was to generate data sets similar to human haplotype data. The simulator described in Section 2.4 was the primary simulator for creating these

data sets. Another option that was considered as an alternative was COSI [104]. It was decided not to use COSI, however, for the following reasons. Applying COSI to generate inversions of varying age would have been cumbersome. While it models varying recombination rates and recombination hotspots and is calibrated to produce data similar to real human SNP data, it does not model double recombinations or the peculiarities of inversions described in Section 1.2. Double recombinations are rare and their exclusion is not a major inaccuracy, especially when the simulated segments are at most 500 kb, but the suppression of recombination in inversion regions leaves a notable mark in the LD patterns in the two subpopulations. Simulating inversions by COSI would have entailed creating a subpopulation with an exponential growth model that had been created from the main population lines several generations ago.

A more relevant limitation is that COSI cannot simulate a segment that is not completely contained within an inversion and as such, cannot be used to generate data for comparing different inversion detection algorithms. Some of the experiments include tests where the simulated inversion does not span the whole simulated segment. To keep all ROC curves comparable, InvCoal was used to do all simulations. This admittedly hurts the accuracy of the ROC curves as estimates for the curves on real data, as InvCoal cannot handle different population histories and varying recombination rates like COSI. As another downside, the ancestral-type effective population size in InvCoal is fixed to a constant-sized population and the inversion-type population undergoes exponential population growth.

This choice of simulator limits the selection of population growth models. The exponential growth model for the inversion population is not realistic and the simulator also does not simulate selection.

Let us consider the chosen population parameters more closely. The ancestral-type effective population size was chosen as 7,500 individuals. There are multiple different estimates for past population sizes. The calibrated parameters of COSI given in [104] used 12,500 to model the effective population before the simulated African expansion 17,000 generations ago. This is higher than a number of other used values for effective population sizes for humans in the past (e.g., [125] with a value of 10,000 and [126] with

estimates 7,500 and 3,100 for different HapMap populations). As a middle-of-the-road option, we chose 7,500.

The population history used by COSI include bottlenecks, but also a considerable increase in the effective population size 200 to 400 generations before the present. The exponential growth in InvCoal very crudely approximates the latter, whereas the former features are not modelled. Unfortunately, the exponential increase in the simulated population size in InvCoal affects only the inversion-type population. This is unrealistic.

To use InvCoal, multiple parameters had to be specified for the simulation. They are summarized in Table 5.3. The gene conversion parameters were selected based on the calibrated parameters for COSI [104]: the initiation probability was taken directly, but the tract length was chosen so that the length had the same expectation as the constant length in the calibrated parameters of COSI, 0.5 kb.

As the interference parameter for the Counting model we chose $m = 4$. Broman and Weber [15] estimated the Gamma model parameter ν to be 4.3, which corresponds to $m = 3.3$ if non-integer values were permitted. Lin and Speed [76] report $m = 4$ to be the best positive integer value for the Counting model in humans.

In the cases where the inversion was supposed to be as long as the simulated segment, the inversion length was set to be only nearly that, i.e., the inversion was at most two basepairs shorter than the segment. This was due to InvCoal's incapability of simulating inversions that spanned the whole simulated segment.

The smallest and largest used ages for inversion haplotypes are rather extreme. The oldest inversions at 150,000 generations are roughly 3 million years old, assuming one generation corresponds to 20 years. The youngest at 5,000 generations are with the same assumption 100,000 years old. Most of the results, however, are given with inversions of age 20,000 or 40,000 generations, to remove focus from cases in which the inversion would likely have been fixed in the population by then.

The HapMap project estimated recombination rates between SNPs [129]. These estimates were used to form a recombination rate distribution for the human genome. Even though the recombination rate in InvCoal is fixed within each simulation round, it is possible to sample the segment recombination rate from this distri-

Table 5.3: Basic parameters used for InvCoal in experiments.

Parameter	Values
Inversion proportion f	$10\%, 20\%, 30\%$
Inversion age	5,000; 20,000; 40,000; 80,000; 150,000 generations
Segment length	150 kb, 250 kb, 500 kb
Inversion length	50 kb, 150 kb, ca 250 kb, ca 500 kb
N_e^{A}	7,500
$N_e^{\mathrm{I}}(0)$	$f \times N_e^{\mathrm{A}}(0)/(1-f)$
Chiasma interference parameter m	4
Mutation rate μ	10^{-8}
Recombination rate r	$10^{-8}, 10^{-9}, 1.3102 \times 10^{-8}$, sampled from the estimated distribution
Gene conversion initiation probability	4.5×10^{-9} per bp
Gene conversion tract length parameter	500 (0.5 kb expected length)

bution of estimated rates provided by the HapMap project and use it to simulate a segment with this value. By computing statistics from these generated data sets and then averaging them, one can expect to gain a reasonable estimate for the mean of the statistic over all inversions in the human genome with the additional assumption of the presence of inversions being independent of recombination rate, i.e., the 'unsuppressed' recombination rates within inversions are the same as elsewhere in the genome. Note that the recombination rate parameter r in simulations translates to the intended recombination rate before the suppression effect is applied.

The recombination rate distribution was computed as follows from the genetic distances computed from Phase II HapMap, release 22 (NCBI build 36). For each chromosome, the first SNP was marked. Then the next SNP to follow it at least 500 kb ahead was also marked, and this was repeated until the whole chromosome was processed. Next, the genetic and physical distances between these marked SNPs were computed. The data were hence now a

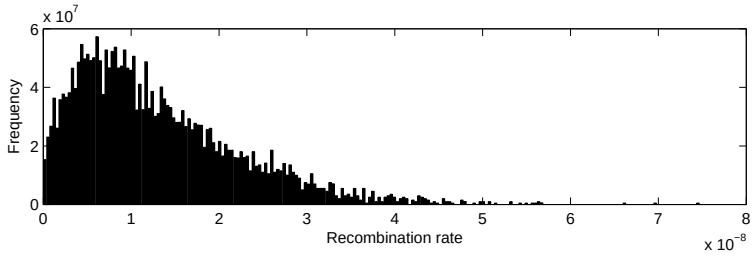


Figure 5.1: Estimated recombination rate histogram from which recombination rates were sampled for InvCoal. The y-axis represents the sum of the lengths of the windows in the bin.

set of pairs of physical and genetic distances from each marked SNP to the next marked SNP. Next, the intervals in which the physical distance was greater than 2 Mb were discarded. This is because NBS-Scan cannot use SNP-free regions. Finally, a histogram for genetic distances per basepair (computed as the genetic distance between SNPs divided by the corresponding physical distance) in this model was created with 214 bins, and used as the recombination rate distribution. Each window was weighted with its length in basepairs. The histogram is depicted in Figure 5.1. The mean of the histogram was 1.3102×10^{-8}. This rate was also used in experiments.

Note that the results from the simulations cannot reliably be used to infer the performance of NBS or other compared methods on real data. This is shown in Section 5.2.4.

5.2.3 Analysis of synthetic inversions

To analyse the performance of NBS under controlled conditions, InvCoal was used to produce synthetic data sets. For each positive scenario (data sets with an inversion) involving NBS alone, 1,000 data sets were generated. For the negative scenario (data sets without an inversion) involving only NBS, 3,000 data sets were generated. Note that the experiments did not use NBS-Scan to decide on the inversion status but NBS alone, with the exception of some tests using R_M to help in determining segment inversion status. In particular, no window joining was done after computing NBS.

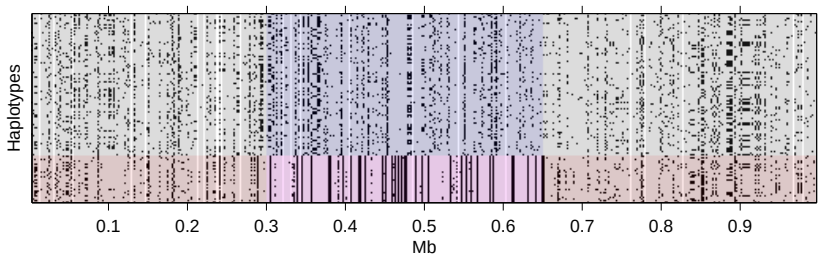


Figure 5.2: An example of an inversion simulated by InvCoal. Inversion-type haplotypes are shaded red and the inversion region is shaded blue. Note the clear bipartition in the inversion region and how it disappears elsewhere.

It is relevant to note that in the case of InvCoal, the results are not truly indicative of how well NBS can detect inversions in humans. This is due to the simulated ancestral history: the model used in the simulations does not completely reflect the believed human population subdivision and migration history. In particular, the effective population size is set to only one estimated value, and this estimate does not necessarily reflect the ancient population size tens of thousands of generations into the past.

Figure 5.2 displays an example of InvCoal output. In this and other similar figures, the covered region is split into bins of equivalent size. If a bin has SNPs in it, one of them is displayed in the plot. Otherwise, the bin is represented by a blank column. This is done to keep the maximum number of displayed SNPs per basepair constant and the number of SNPs manageable. The SNPs were omitted only from the plot and not from the computation.

The power of inversion detection

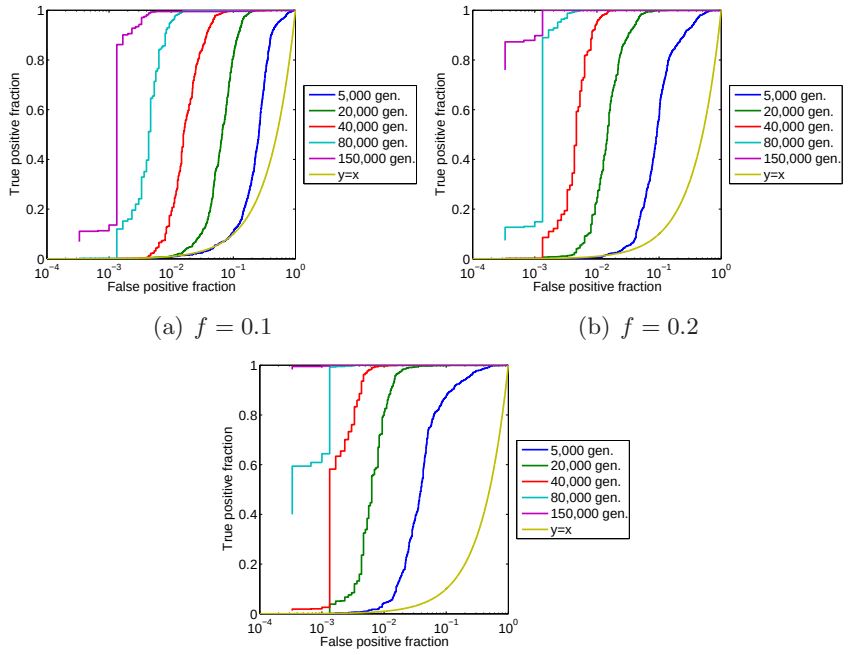
In all panel-ascertained SNP experiments, mean SNP density was one per 2 kb.

In the first round of InvCoal experiments, different recombination parameters (r was set to 10^{-8}, 10^{-9}, 1.3102×10^{-8} or r was sampled from the recombination rate histogram) were used while the simulated segment length, together with that of the inversion, was 250 kb or 500 kb. The SNP ascertainment method was the sim-

ulated 48 haplotype panel method. A simple miscall error mask was applied; each measurement was miscalled with probability 0.001. These errors were independent and identically distributed. SNPs with MAF below 0.05 were ignored. Hudson and Kaplan's R_M (an estimate for the minimum number of recombinations needed to produce the observed SNP data) [58] was not used for filtering in an attempt to eliminate recombination cold spots as false positives; its effect is discussed later in this section.

In the used experiment configurations, older inversions were more reliably detected than younger ones (Figure 5.3) in all configurations. This is surprising, because young inversions in the used population model were expected to quickly grow into very homogeneous haplotype blocks. One would therefore assume that such inversions would have been easier to detect. One possible reason is that with older inversions there were more mutations that appeared after the inversion event in the inversion population, thus increasing the proportion of mutations of which the novel allele was limited to the inversion population alone. In particular, if the MRCA within the inversion population was found much earlier than the actual inversion event, as is likely in the case of old inversions, then there were a considerable number of SNPs whose alleles directly corresponded to the inversion status of the haplotypes. This makes the separation between the two arrangements clearer. As a comparison, Stefansson et al. [117] estimated the age of the 900-kb inversion in chromosome 17 to be 3 million years, which would mean 150,000 generations, if each generation is assumed to last 20 years. The SNP data from the inversion region is shown in Figure 3.2 after sorting the haplotypes and SNPs conveniently. This inversion strongly displays the four-field pattern, which supports the previously mentioned theory. However, the results of Donnelly et al. [30] suggest the MRCA of the inversion is actually $656.8 - 1313.6$ or $2167.4 - 4334.7$ generations old. This undermines the experiment setup for synthetic inversions that were all at least 5,000 generations old. The relevance of inversion ages is discussed later in this chapter.

Naturally, as seen in Figure 5.3, the higher the inversion frequency, the easier it was to detect their presence. It appears that in cases where inversion frequency was 0.1, NBS could not provide reliable results. Because NBS detects the signal arising from the bipartition of haplotypes, it is not expected to work well if either



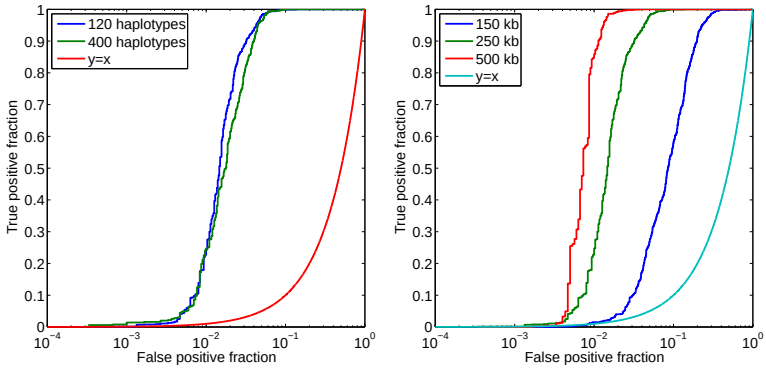
(a) $f = 0.1$ (b) $f = 0.2$

(c) $f = 0.3$

Figure 5.3: NBS ROC curves computed from InvCoal-generated data sets with different inversion ages, ranging from 5,000 generations to 150,000 generations and using three different inversion frequencies f. The recombination rate was sampled from the histogram. The data sets had 120 haplotypes and the window size was 250 kb.

subpopulation is very small. In this aspect, it is inferior to the endpoint signal detecting methods [9, 112] when the alternative-type arrangement is the more common type.

Increasing the number of haplotypes had practically no effect on ROC-curves (Figure 5.4(a)). The increase in window size (Figure 5.4(b)) notably improved on the performance of NBS. On the first glance, this is not surprising. But each 250-kb-long window had 125 SNPs, which should be a sufficient amount for the spectral ordering (see Section 3.3) to find only a good ordering of the SNPs, if there was one. Because NBS treats every SNP independently and more SNPs only make the SNP-wise compression ratio less sensitive



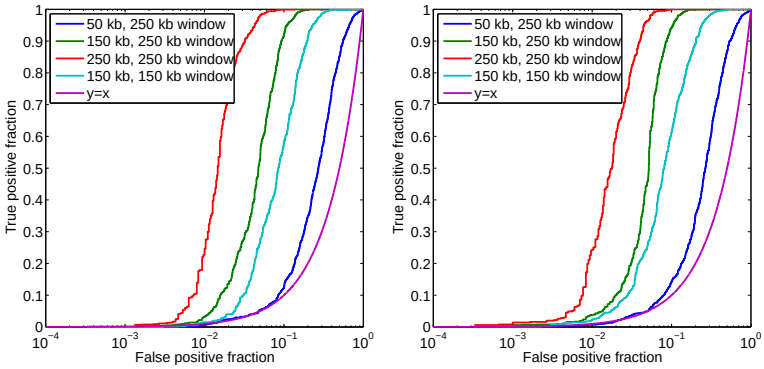
(a) 250-kb window (b) 120 haplotypes

Figure 5.4: NBS ROC curves for InvCoal-generated data sets with different sizes in terms of haplotypes and length of the window. The recombination rate was sampled from the histogram. The inversion age was 20,000 generations and inversion frequency f was 0.2.

to random noise, adding more SNPs should not improve the score's performance. However, a longer window means also that even in regions of low recombination rate there are more recombinations to suppress in heterokaryotypes, thus strengthening the subpopulation division signal in comparison to the null hypothesis scenario. Hence, the effect is possibly due to the increased genetic length of the segment rather than the number of SNPs.

Note that even though the length of the simulated inversion also varies with the window size, the results for scoring a 250-kb window of a 500-kb inversion would be practically the same as scoring a 250-kb inversion. The largest difference in the scenarios would be due to the double recombinations, but in both cases they are very rare. Furthermore, because gene conversion rates were considered in the simulation to be equal over the whole inversion regardless of the distance to the end-points, the segments generated under these different conditions would appear similar with respect to gene conversions.

In the second round of experiments, InvCoal was used to produce inversions that spanned 50 kb or 150 kb of the 250-kb simulated segment; the inversions were placed following a uniform distribution



(a) $f = 0.2$, 120 haplotypes (b) $f = 0.2$, 400 haplotypes

Figure 5.5: NBS ROC curves showing the power to detect inversions of different length, based on InvCoal-generated data sets. Inversion age is 20,000 generations.

so that they were completely contained within the segment. The recombination rate was sampled from the recombination rate histogram. Data sets with a 150-kb inversion covering all of a 150-kb window were also simulated.

The effect of the relative inversion length (relative to the window size) is clear (Figure 5.5). The drop from a window-sized inversion to a 150-kb-long inversion is slightly surprising, because the population subdivision in InvCoal output remains for some distance outside the inversion end-points, which could have caused the precise window size to have only a small effect. It is possible, however, that this does not hold in simulations where the recombination rate varies within the segment.

The effect of the varying recombination rate is noteworthy (Figure 5.6). Because NBS detects the suppressed recombination between arrangements, the more recombinations there are in the null scenario, the stronger the difference between the inversion and null case simulations is. This contributes to the very promising ROC curve for case $r = 10^{-8}$. However, if the recombination rate is considerably lower ($r = 10^{-9}$), the power is much smaller. In a way, the recombination rate is a limit to the strength of the signal NBS detects. The variance in the recombination rate across the

genome also strongly affects the performance of NBS. The mean of the estimated recombination rate histogram in Figure 5.1 was 1.3102×10^{-8}, and the ROC curve for that was located above the curve for the case $r = 10^{-8}$ in the figure. For this reason, the curve of $r = 1.3102 \times 10^{-8}$ was excluded from the figure. Yet, as seen in Figure 5.6, in the more realistic scenario where the recombination rate varied according to the estimated distribution, the curve is considerably lower.

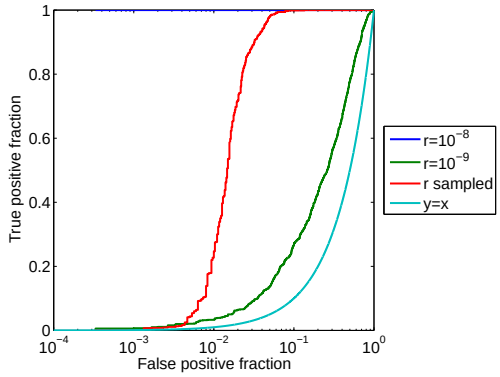
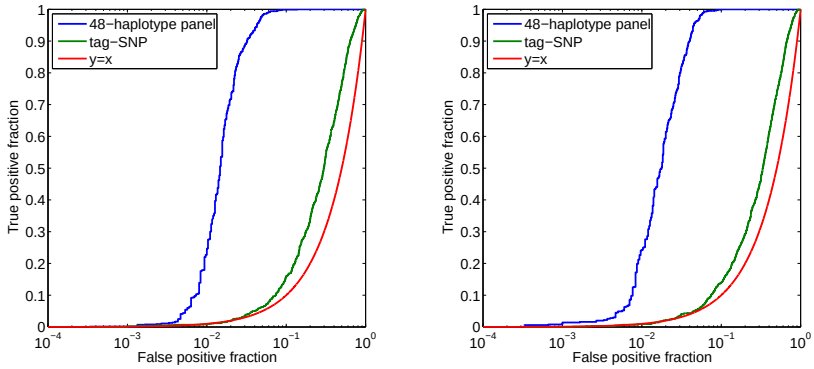


Figure 5.6: NBS ROC curves for the cases with different values for the recombination rate r: either a constant of 10^{-8} or 10^{-9} or sampled from the histogram of Fig. 5.1. Inversion frequency is 0.2, window size is 250 kb, inversion age 20,000 generations and there are 120 haplotypes.

The full-window inversion experiments were done also using the adapted tag-SNP algorithm of Carlson et al. [17]. There was a notable difference in the ROC curves between the two ascertainment schemes (Figure 5.7).

The reason for the large difference is not obvious. One possible explanation is that NBS detects SNPs that have a particular kind of high LD between them. The tagSNP-selection algorithm of Carlson et al. [17] does not distinguish between high LD and high LD suitable for NBS to detect, so the algorithm effectively eliminates the signal NBS is attempting to detect. It is possible that using LD-based tag-SNP selection algorithms in general would be detrimental for NBS's applicability, but further experiments on multiple SNP-



(a) $f = 0.2$, 120 haplotypes, 250 kb window

(b) $f = 0.2$, 400 haplotypes, 250 kb window

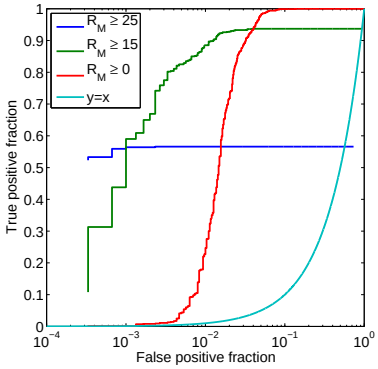
Figure 5.7: NBS ROC curves for cases where the SNP selection scheme varies in InvCoal-generated data sets. The inversion age is 20,000 generations.

selection algorithms would be necessary to determine which aspects of the SNP selection affect the results the most. While our tag-SNP algorithm implementation does produce data sets with fewer SNPs than the panel simulation described in Section 5.2.1, the data sets had mean SNP spacing under 7 kb.

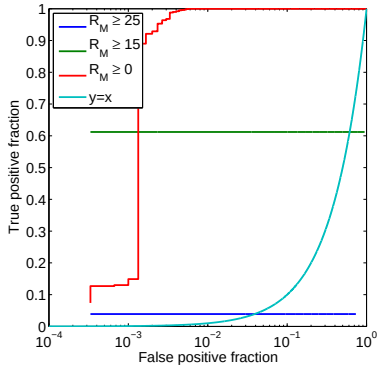
The effect of using recombination measure filtering discussed in Section 3.4 on the results is shown in Figure 5.8. Based on the examination of the results, while the filtering seems to decrease the number of false positives with high values of NBS, it also eliminates a considerable proportion of true positives. The inversion frequency also affects the measure. The more frequent the inversion is, the smaller the R_M threshold should be to gain the same effect.

However, recall that the used simulator InvCoal does not model varying recombination rates within the simulated segment. This also means that the simulator does not accurately model recombination cold spots; since this causes the recombinations to be evenly spread on the simulated segment, R_M possibly gets higher values than when the recombinations are concentrated in recombination hot spots. Hence, the best R_M threshold values are likely to be different for real data.

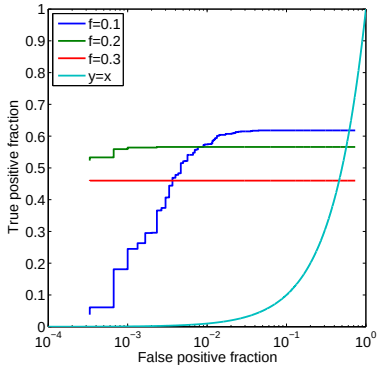
In summary, NBS would seem to perform well when both sub-



(a) $f = 0.2$, 120 haplotypes, 250 kb window, inversion age 20,000 generations



(b) $f = 0.2$, 120 haplotypes, 250 kb window, inversion age 80,000 generations



(c) 120 haplotypes, 250 kb window, inversion age 20,000 generations, max. subpopulation $R_M \geq 25$

Figure 5.8: NBS ROC curves for different levels of R_M thresholds and simulation parameters in InvCoal-generated data sets.

populations are represented in the data set with at least 20% of the haplotypes and the window is at least 250 kb long. The age of the inversion is an important factor: the older the inversion, the more reliably it is detected from SNP data sets. By comparison, increasing the number of haplotypes in the data set does not notably increase the performance of NBS.

Comparison of inversion-detection algorithms

To compare both NBS and Sindi and Raphael's method [112], the experiment setup had to be changed. For a brief description of the latter, see Section 3.5. This method is called SR-method in this thesis. The reason why the setup is different is that whereas NBS detects the presence of an inversion from within, SR-method detects the signal left at the end-points and requires SNPs both inside and outside the inversion.

With this in mind, InvCoal was used to generate 750-kb segments with 120 haplotypes with the recombination rates sampled from the histogram. The inversion size was 500 kb and it was placed randomly along the segment so that at least 50 kb of non-inverted material was left at both ends.

NBS used 250-kb windows and 50-kb jumps. In this case, the score for the complete segment was the maximum value of NBS observed in the investigated windows. In particular, windows were not joined together and then re-evaluated. The window also did not go beyond the end of the segment, i.e., the last 250-kb window that was considered started at the 500 kb mark. Had the partial windows been included, NBS would have performed notably worse.

In the case of SR-method, the segment was given the smallest empirical score, measured from the empirical distribution of likelihood ratios as described by Sindi and Raphael [112], from all putative end-point pairs that were at least 200 kb apart. If no potential pair of end-points in the data set was observed, the data set was assigned an empirical score of 1. We address the generation of the entropy distributions and the empirical likelihood test ratio scores in a moment.

Also the hybrid method, where NBS was utilized in selecting putative end-point pairs, was investigated in this experiment setup. In this case, the threshold for including a gap between SNPs was

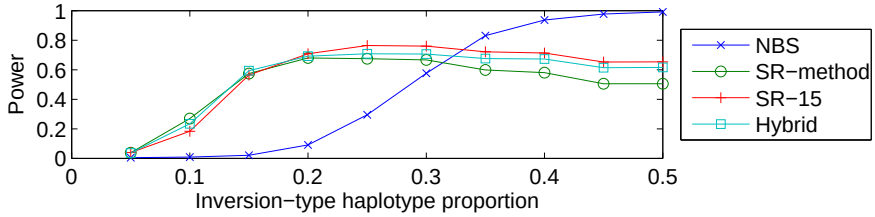
0.09.

The third compared method, labeled SR-15, was the same as SR-method with the exception that the SNP gap was chosen as a potential endpoint if the entropy was in the top 15% instead of top 10%. The reason for its inclusion was to investigate how much including more potential end-points in consideration affects the performance of Sindi and Raphael's method.

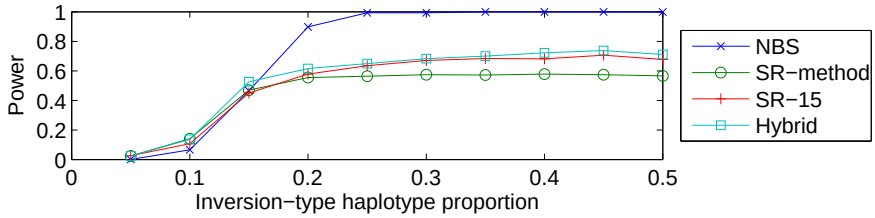
Let us now take a closer look at how SR-method, SR-15 and the hybrid method were adjusted to work with synthetic data sets. This is an adaptation of how Sindi and Raphael [112] use the SR-method for real data sets. First, InvCoal was used to produce 2,000 data sets with 120 haplotypes in each, each data set representing the SNPs in a chromosome segment of 1 Mb with no inversion present and the recombination rate for each data set sampled from the histogram in Figure 5.1. The SNPs underwent simulated panel ascertainment and removal of SNPs with MAF below 0.05. From these resulting data sets, the entropy distributions for different window widths L ranging from 3 to 15 were computed for use by SR-method, SR-15 and the hybrid method.

The next step was to simulate another 2,000 data sets without inversions. These underwent the same filtering steps as the ones used to generate the entropy histograms, but in this case, these data sets were used to create the empirical distribution for the test score. The potential inversion end-points were first chosen according to the method for which the empirical score distribution was to be constructed. After that, the EM-algorithm of Sindi and Raphael [112] was used to compute the test statistic for all pairs of end-points that could be useful in determining the empirical scores for the simulations with inversions. The likelihood ratio test score, the distance between end-points and the degrees of freedom were recorded for each observation. This resulted in empirical score distributions for each of the three EM-based methods separately. Finally, these distributions were used to score the observed likelihood test ratio scores in the actual simulations used to create power curves.

Figure 5.9 displays the power (proportion of true positives to all positives) of these different inversion detection methods when the ancestral-type population is the standard order. By comparison, Figure 5.10 displays the case where the ancestral-type population is the alternative order. Because the simulator uses the ancestral



(a) Inversion age 20,000 generations



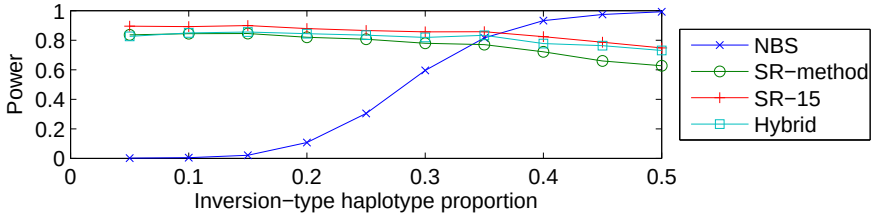
(b) Inversion age 40,000 generations

Figure 5.9: The power of inversion-detection schemes with a fixed false positive fraction of 0.01. The ancestral-type haplotypes are the standard-type haplotypes.

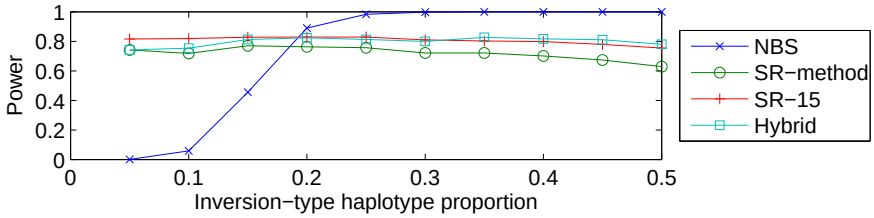
order as the reference order, in these latter simulations the majority of the haplotypes was reversed with respect to the reference order. In both figures, the false positive fraction was fixed at 0.01. There were 1,000 positive simulations for each parameter configuration and 3,000 null simulations.

Note that the cases where the inversion frequency is high are not realistic scenarios due to the way the inversion-type population size was computed. At the extreme in the case of $f = 0.9$, the inversion-type effective population size would be 67,500. These values should not be considered as indicative of the presence of the power for a method. The values of $f = 0.4$ or $f = 0.5$ are already slightly tending towards unrealistic scenarios.

As mentioned, SR-method inspects only such breakpoints where the entropy of the haplotypes around it is in the top 10% in the genome for that window size. The inversion simulations, however, produce data sets where the haplotype diversity is low within the inversion and at the end-points. This resulted in notably fewer end-points being considered as potential inversion end-points, and in some cases, there were no two proposed end-points within the



(a) Inversion age 20,000 generations



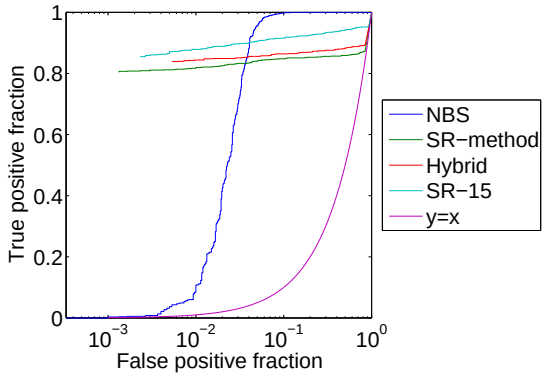
(b) Inversion age 40,000 generations

Figure 5.10: The power of inversion-detection schemes with a fixed false positive fraction of 0.01. The inversion-type haplotypes are the standard-type haplotypes.

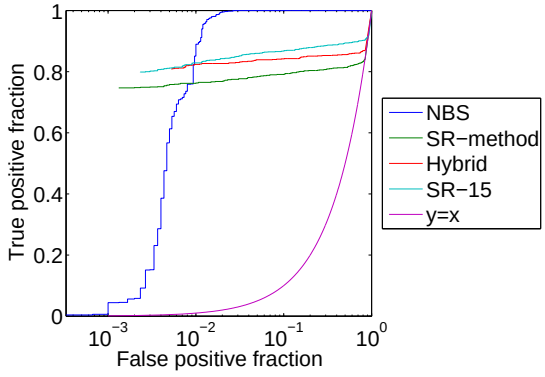
simulated segment. This was computed as a empirical score of 1 for the data set. By comparison, in the cases where there were an inversion and multiple potential end-points present, the empirical score was in many cases 0. One possible explanation for the non-varied haplotype structure is the used inversion population growth model, which affects also the regions outside the inversion.

This results in the depicted SR-method power curves being similar to the case where the false positive fraction is 0.1. By comparison, the power of NBS-Scan increases notably compared to the case of fixing the false positive fraction to 0.01. This is shown in Figure 5.11, which shows the ROC curve for the case of inversion frequency being 0.2 and inversion age being 20,000 generations. The figures therefore depict the ROC curves underlying one point in Figures 5.10(a) and 5.10(b). The figure also shows that if the false positive fraction is lowered, the performance of NBS-Scan decreases sharply whereas the power of SR-method remains at nearly the same levels.

We note that the hybrid method is not noticeably better than the original method of Sindi and Raphael. SR-15-variant, however,



(a) Inversion age 20,000 generations



(b) Inversion age 40,000 generations

Figure 5.11: ROC curves for inversion-detection schemes with the inversion frequency $f = 0.2$. The inversion-type haplotypes are the standard-type haplotypes.

is better by a considerable margin when the inversion is not rare. This suggests that less strict criteria for potential inversion end-points might be useful. It is likely that the biggest reason for the loss of power for the SR-method variants in these experiments on synthetic data is that the real inversion end-points are not considered as potential inversion end-points. Nonetheless, because the result is based on simulations that are only remotely indicative, an inspection into the performance of the two methods on real data is more useful. This is briefly addressed in Section 5.2.4.

As can be expected, NBS is indifferent to which haplotype population is actually the reference sequence. However, these experiments did not address the question of what happens when the ancestral-type population is the rarer type and the inversion-type population is the more common one.

5.2.4 Real data sets

Some of the HapMap Phase III (release 2, February 2009) data sets discussed in Section 1.4 were processed in 250-kb and 500-kb windows starting at 50 kb intervals. As a preprocessing step, SNPs with MAF ≤ 0.05 were removed, as SNPs with low MAF are at best uninformative to NBS. This was also likely to have the result of excluding inversions with MAF below that threshold out of the search. The data sets were in NCBI build 36 coordinates; these physical coordinates will be used throughout this section on inversions in real-world data sets.

The studies were limited to autosomes because of the limited amount of recombination the sex chromosomes undergo. While the YRI and CEU data sets in phase III contained not only trios but also duos and unrelated individuals, in these experiments only the haplotypes resolved by phasing trios were used. This means that the used data sets have fewer haplotypes than were provided on the HapMap website. If we look at Figure 5.4(a), we see that the effect of the additional haplotypes would likely not have been large. The YRI and CEU data sets used in this section are therefore only trios, whereas CHB and JPT data are unrelated individuals. The used data sets contained phased haplotypes, so the difference comes in CHB and JPT data being less reliably phased. It is possible that the results are slightly affected by phasing errors. In total, the CEU data set had 1,075,275 SNPs, the YRI data set 1,142,161 and the

JPT+CHB data set 938,868 SNPs.

For a number of different NBS-Scan parameter values, the sample p-value was estimated by a Monte Carlo method. In order, these parameters were the size of the sliding window, the minimum NBS for a region to be labeled an inversion, the minimum number of SNPs required to be within a window for it to be considered and the minimum required ratio of R_M per the number of SNPs within the joint population (see Section 3.4). The p-values estimate whether the candidate set returned with the listed parameters covers significantly more basepairs known to belong into inversions than a randomly assembled candidate set. In each scenario, 100,000 different pseudo-candidate sets were generated, composed of a varying number of regions covering in total at least as many basepairs as the real candidate set but at most L basepairs more, where L is the window size used in the analysis. These regions were generated by sampling uniformly at random over all 22 autosomes the windows that contained at least the specified minimum amount of SNPs.

Then, for each random sample the number of basepairs they covered was computed and the p-value for the parameters was reported the proportion of the random samples with higher number of overlapping basepairs than the real candidate set. Algorithm 5.1 summarizes the process in greater detail.

The p-values for a number of different parameter combinations for joining windows were computed for the results for the JPT+CHB data set. The computed p-values for the three investigated data sets are given in Tables 5.4, 5.5 and 5.6. The entries labeled 'Inversion' in the Database of Genomic Variants[1] [59] (DGV) were used as the set of known inversions. There were in total 825 such entries in July 2009 release for hg18 in autosomes. The known inversions covered in total 44,905,910 basepairs of the autosomes. The computed p-values have not been corrected for multiple testing that resulted from using several different parameter values.

Naturally, because these p-values are based on only a currently known set of inversions present in the human genome, they or the null hypothesis cannot be considered to be invariant over database releases and time. They also may show bias with respect to the populations that have been more thoroughly studied for polymor-

[1] `http://projects.tcag.ca/variation/` (Accessed 02.11.2009)

Table 5.4: Sample coverage p-values for different criteria in selecting permissible windows in the CEU data set inversion region candidates. Each test used 100,000 random windows. No multiple testing correction has been used in reporting the p-values. The line with bold face represents the set of parameters chosen for reporting the candidate lists from the experiments.

Window size (kb)	NBS	#SNPs	$\frac{R_M}{\#\text{SNPs}}$	Coverage (kb)	Overlap (kb)	p
250	≥ 0.5	≥ 10	≥ 0	178,150	4,799	0.00055
250	≥ 0.5	≥ 10	≥ 0.1	143,700	4,527	0.00007
250	≥ 0.5	≥ 20	≥ 0	175,700	3,882	0.01626
250	≥ 0.5	≥ 20	≥ 0.1	140,100	3,773	0.00218
250	≥ 0.5	≥ 50	≥ 0	152,050	3,094	0.05494
250	≥ 0.5	≥ 50	≥ 0.1	114,550	2,307	0.08754
250	≥ 0.6	≥ 10	≥ 0	96,000	3,870	0.00002
250	≥ 0.6	≥ 10	≥ 0.1	56,850	2,584	0.00005
250	≥ 0.6	≥ 20	≥ 0	91,500	3,157	0.00023
250	≥ 0.6	≥ 20	≥ 0.1	52,150	2,011	0.00151
250	≥ 0.6	≥ 50	≥ 0	71,100	1,352	0.18198
250	≥ 0.6	≥ 50	≥ 0.1	36,500	677	0.25735
250	≥ 0.7	≥ 10	≥ 0	40,850	1,494	0.00722
250	≥ 0.7	≥ 10	≥ 0.1	16,950	318	0.28939
250	≥ 0.7	≥ 20	≥ 0	38,900	1,346	0.01348
250	≥ 0.7	≥ 20	≥ 0.1	15,550	318	0.25087
250	≥ 0.7	≥ 50	≥ 0	25,250	603	0.15483
250	≥ 0.7	≥ 50	≥ 0.1	8,950	0	0.84612
500	≥ 0.5	≥ 10	≥ 0	55,850	3,632	0.00002
500	≥ 0.5	≥ 10	≥ 0.1	43,350	3,512	0.00001
500	≥ 0.5	≥ 20	≥ 0	53,700	3,230	0.00015
500	≥ 0.5	≥ 20	≥ 0.1	41,200	3,111	0.00006
500	≥ 0.5	≥ 50	≥ 0	49,700	2,142	0.00801
500	≥ 0.5	≥ 50	≥ 0.1	36,600	2,135	0.00171
500	≥ 0.6	≥ 10	≥ 0	28,050	2,298	0.00019
500	≥ 0.6	≥ 10	≥ 0.1	16,200	2,250	0.00001
500	≥ 0.6	≥ 20	≥ 0	26,000	1,936	0.00079
500	≥ 0.6	≥ 20	≥ 0.1	15,300	1,887	0.00007
500	≥ 0.6	≥ 50	≥ 0	21,750	1,153	0.0181
500	≥ 0.6	≥ 50	≥ 0.1	11,550	1,105	0.00397
500	≥ 0.7	≥ 10	≥ 0	12,650	628	0.05368
500	≥ 0.7	≥ 10	≥ 0.1	5,350	334	0.09546
500	≥ 0.7	≥ 20	≥ 0	12,200	628	0.05109
500	≥ 0.7	≥ 20	≥ 0.1	4,900	334	0.08656
500	≥ 0.7	≥ 50	≥ 0	9,800	794	0.01581
500	≥ 0.7	≥ 50	≥ 0.1	2,400	0	0.36597

Table 5.5: Sample coverage p-values for different criteria in selecting permissible windows in the YRI data set inversion region candidates. Each test used 100,000 random windows. No multiple testing correction has been used in reporting the p-values. The line with bold face represents the set of parameters chosen for reporting the candidate lists from the experiments.

Window size (kb)	NBS	#SNPs	$\frac{R_M}{\#\text{SNPs}}$	Coverage (kb)	Overlap (kb)	p
250	≥ 0.5	≥ 10	≥ 0	38,150	1,140	0.0387
250	≥ 0.5	≥ 10	≥ 0.1	30,550	782	0.10936
250	≥ 0.5	≥ 20	≥ 0	37,800	1,429	0.00703
250	≥ 0.5	≥ 20	≥ 0.1	29,600	748	0.11688
250	≥ 0.5	≥ 50	≥ 0	23,750	442	0.28522
250	≥ 0.5	≥ 50	≥ 0.1	18,100	133	0.62758
250	≥ 0.6	≥ 10	≥ 0	11,900	2	0.86285
250	≥ 0.6	≥ 10	≥ 0.1	7,700	2	0.71896
250	≥ 0.6	≥ 20	≥ 0	11,650	325	0.15363
250	≥ 0.6	≥ 20	≥ 0.1	6,450	2	0.65058
250	≥ 0.6	≥ 50	≥ 0	6,950	3	0.66413
250	≥ 0.6	≥ 50	≥ 0.1	3,750	2	0.47259
250	≥ 0.7	≥ 10	≥ 0	2,950	2	0.38123
250	≥ 0.7	≥ 10	≥ 0.1	1,700	2	0.24387
250	≥ 0.7	≥ 20	≥ 0	2,850	2	0.38188
250	≥ 0.7	≥ 20	≥ 0.1	1,750	2	0.2731
250	≥ 0.7	≥ 50	≥ 0	1,700	2	0.24176
250	≥ 0.7	≥ 50	≥ 0.1	1,100	2	0.17993
500	≥ 0.5	≥ 10	≥ 0	19,050	780	0.06573
500	≥ 0.5	≥ 10	≥ 0.1	15,450	454	0.21901
500	≥ 0.5	≥ 20	≥ 0	16,700	481	0.23069
500	≥ 0.5	≥ 20	≥ 0.1	13,250	155	0.36774
500	≥ 0.5	≥ 50	≥ 0	14,350	758	0.04016
500	≥ 0.5	≥ 50	≥ 0.1	10,850	347	0.18715
500	≥ 0.6	≥ 10	≥ 0	5,100	291	0.10266
500	≥ 0.6	≥ 10	≥ 0.1	3,300	2	0.3762
500	≥ 0.6	≥ 20	≥ 0	4,000	291	0.08336
500	≥ 0.6	≥ 20	≥ 0.1	2,200	2	0.28667
500	≥ 0.6	≥ 50	≥ 0	2,750	375	0.04391
500	≥ 0.6	≥ 50	≥ 0.1	900	2	0.12429
500	≥ 0.7	≥ 10	≥ 0	400	0	0.08671
500	≥ 0.7	≥ 10	≥ 0.1	400	0	0.0889
500	≥ 0.7	≥ 20	≥ 0	400	0	0.08672
500	≥ 0.7	≥ 20	≥ 0.1	400	0	0.08887
500	≥ 0.7	≥ 50	≥ 0	0	0	–
500	≥ 0.7	≥ 50	≥ 0.1	0	0	–

Table 5.6: Sample coverage p-values for different criteria in selecting permissible windows in the JPT+CHB data set inversion region candidates. Each test used 100,000 random windows. No multiple testing correction has been used in reporting the p-values. The line with bold face represents the set of parameters chosen for reporting the candidate lists from the experiments.

Window size (kb)	NBS	#SNPs	$\frac{R_M}{\#\text{SNPs}}$	Coverage (kb)	Overlap (kb)	p
250	≥ 0.5	≥ 10	≥ 0	214,000	6,433	$\leq 10^{-5}$
250	≥ 0.5	≥ 10	≥ 0.1	202,700	5,912	0.00005
250	≥ 0.5	≥ 20	≥ 0	208,000	5,805	0.00012
250	≥ 0.5	≥ 20	≥ 0.1	196,600	5,266	0.00032
250	≥ 0.5	≥ 50	≥ 0	174,550	3,775	0.02159
250	≥ 0.5	≥ 50	≥ 0.1	159,950	3,370	0.03556
250	≥ 0.6	≥ 10	≥ 0	134,400	3,766	0.00138
250	≥ 0.6	≥ 10	≥ 0.1	111,250	3,449	0.0006
250	≥ 0.6	≥ 20	≥ 0	125,850	3,241	0.00608
250	≥ 0.6	≥ 20	≥ 0.1	102,550	2,320	0.04217
250	≥ 0.6	≥ 50	≥ 0	86,000	1,177	0.47656
250	≥ 0.6	≥ 50	≥ 0.1	69,550	1,144	0.30759
250	≥ 0.7	≥ 10	≥ 0	69,700	1,913	0.01941
250	≥ 0.7	≥ 10	≥ 0.1	48,850	1,258	0.05996
250	≥ 0.7	≥ 20	≥ 0	64,400	1,234	0.18749
250	≥ 0.7	≥ 20	≥ 0.1	43,750	323	0.79962
250	≥ 0.7	≥ 50	≥ 0	35,100	317	0.68069
250	≥ 0.7	≥ 50	≥ 0.1	22,850	17	0.93898
500	≥ 0.5	≥ 10	≥ 0	65,000	3,236	0.00044
500	≥ 0.5	≥ 10	≥ 0.1	63,000	3,236	0.00043
500	≥ 0.5	≥ 20	≥ 0	64,250	2,729	0.00332
500	≥ 0.5	≥ 20	≥ 0.1	62,100	2,729	0.00291
500	≥ 0.5	≥ 50	≥ 0	61,600	1,931	0.0387
500	≥ 0.5	≥ 50	≥ 0.1	59,050	1,931	0.03312
500	≥ 0.6	≥ 10	≥ 0	45,650	2,483	0.00119
500	≥ 0.6	≥ 10	≥ 0.1	42,150	2,183	0.00285
500	≥ 0.6	≥ 20	≥ 0	42,400	1,370	0.06056
500	≥ 0.6	≥ 20	≥ 0.1	39,350	1,336	0.05355
500	≥ 0.6	≥ 50	≥ 0	37,050	423	0.53961
500	≥ 0.6	≥ 50	≥ 0.1	33,700	419	0.50056
500	≥ 0.7	≥ 10	≥ 0	19,500	1,790	0.00046
500	≥ 0.7	≥ 10	≥ 0.1	18,050	1,491	0.00229
500	≥ 0.7	≥ 20	≥ 0	17,400	1,002	0.02444
500	≥ 0.7	≥ 20	≥ 0.1	16,250	1,002	0.02165
500	≥ 0.7	≥ 50	≥ 0	12,250	445	0.1733
500	≥ 0.7	≥ 50	≥ 0.1	11,700	445	0.16583

Algorithm 5.1 Computation of p-value for given set of analysis parameters.

Input: The candidate set S, number of samples *repeats*, window size.

Output: Probability p that at least the same number of known basepairs within inversions would be covered by chance.

1: $c_S \leftarrow$ the number of basepairs covered by S (coverage).
2: $p \leftarrow 0$
3: $o_S \leftarrow$ the size of the intersection of S and known inversions.
4: **for** $i = 1, \ldots,$*repeats* **do**
5: Random sample coverage $c_R \leftarrow 0$.
6: Remove all windows from random sample R.
7: **while** $c_R \leq c_S$ **do**
8: Add one window to R chosen at random from the set of all suitable windows. The set of suitable windows are represented by the starting points between every 50 kb that have at least 10 SNPs. The window sizes are adjusted to eliminate SNP-free regions at both ends in steps of 50 kb.
9: Recompute c_R.
10: **end while**
11: $o_R \leftarrow$ the size of the intersection of R and known inversions.
12: **if** $o_R > o_S$ **then**
13: $p \leftarrow p + 1/repeats$
14: **end if**
15: **end for**

phisms. This means that inversions that are present only in one population in high numbers should not be expected to be detected also in the other populations. Yet, this is what this simple test does.

However, because the overlap is computed per basepair, long inversions have more weight in the p-value computation than short ones. This is in the favour of NBS-Scan results in the sense that NBS-Scan can get good p-values with such an evaluation, because long inversions are expected to be detected more reliably than short ones.

Note that already 250-kb windows give sufficiently good p-values so that the result is that the algorithm is believed to perform better

than a random guess. This has a number of potential explanations. First, the small window size may permit the signal of short inversions to show through better than in the case where the window size is twice as long. Second, in the case of synthetic data we used independent windows. In reality, however, the windows are not independent but overlap, and joining together multiple windows in a haplotype block region may very well produce regions where smaller, individual windows may have high scores, but after joining the windows together, the union no longer has as clear a division into two sets of haplotypes and is discarded. Recall that in Algorithm 3.1 we eliminate such combined windows with NBS below the threshold. In effect, the joining of the windows effectively functions as if we performed the investigation on windows larger than the original 250 kb.

The p-values in Tables 5.4, 5.5 and 5.6 were surprising, as increasing the required number of SNPs present inside the window strongly decreased the performance of NBS-Scan. For example, see the rows of Table 5.4 where the number of SNPs was required to be at least 50.

Partly based on these scores, the parameter configuration of window size 250 kb, NBS at least 0.5, minimum number of SNPs 20 and no recombination count filtering was chosen for reporting candidate regions in the data sets. Of the tested parameter combinations, this gives the smallest arithmetic mean of p-values over the three data sets and the smallest maximum of the p-values in the data sets. It also produces interesting precision-recall curves, which are discussed later in this section. However, the geometric mean of the p-values over the three data sets is not minimized by the chosen set of parameters. It is apparent that permitting windows with low number of SNPs allowed for more basepairs within inversions to be found in the CEU and JPT+CHB data sets. This was an interesting phenomenon, because prior expectations said that higher number of SNPs would be useful because small number of SNPs can be situated in a short span of the window. Because these SNPs would then be in high LD, the SNPs are more likely to have a high NBS score. It is possible that regions of low genotyped SNP density occur more frequently with inversions, resulting in this bias.

Based on Tables 5.4, 5.5 and 5.6, it appears that using recombination measures to distinguish inversions from haplotype blocks

as outlined in Section 3.4 is not an effective approach for all cases, as in at least half of the investigated parameter combinations the p-values are larger with the filtering. Hence the results have not been filtered by R_M thresholds; the observed ratios of R_M/SNPs are still given, though, for optional removal of such candidates. Because each successive pair of SNPs can increase R_M only by 1 or 0, normalizing R_M with the number of SNPs is a sensible option, and this was done in Tables 5.4, 5.5 and 5.6. An alternative approach would have been to use the candidate region length to normalize it, but the SNP counts in different regions of similar length could vary significantly.

Candidate lists for HapMap data sets

The top-scoring inversion candidates proposed by NBS-Scan (Algorithm 3.1) have been collected in Tables 5.7, 5.8 and 5.9. In total, there were 506 candidate regions in CEU data set, 123 in YRI data set and 610 in JPT+CHB data set. The reported p-values have been computed for these complete lists. The reference field contains references to publications in which an inversion has been detected intersecting the proposed region by at least one basepair. Note that this also means the real inversion might be a very short one, whereas the inversion candidate is considerably long.

The candidates for all three data sets are listed separately. Inversions occur at different frequency in different data sets, as shown, for instance, by Antonacci et al. [4], who tested for the presence of known inversions in samples from three of the HapMap sets. By listing the best-ranking regions per data set, the differences between different populations become apparent. The NBS histograms also vary between the populations, which makes it difficult to decide on the best-scoring candidate regions in the combined list.

The candidate list for the joint JPT+CHB data set is the longest even though it contained least SNPs. This may be caused by the fact that the data set consists of two smaller data sets, which may have induced an occasionally detectable subdivision in the data set. NBS would falsely notice this as signal for the presence of an inversion. On the other hand, the shortness of YRI candidate set is also surprising. This topic is discussed later in this section.

Some candidate regions were listed in more than one population.

Table 5.7: The first 35 estimated inversions from HapMap data, CEU data set, ordered by NBS score. The window size was 250 kb and each window had at least 20 SNPs. The estimated sample p-value was 0.01626. The data set had 176 haplotypes. The MAF column represents the proportion of the smaller inferred subset. See Section 3.4 for description of R_M. 'Chr.' is the chromosome of the inversion; 'Ref.' gives literature references.

Chr.	Start (Mbp)	Length (Mbp)	NBS	$\frac{R_M}{\#\text{SNPs}}$	MAF	Ref.
1	25.50	0.25	0.796	0.087	0.49	[64]
2	110.20	0.15	0.778	0.114	0.29	[64]
4	26.15	0.25	0.744	0.13	0.44	
17	42.05	0.15	0.743	0.069	0.23	
16	57.80	0.25	0.706	0.264	0.41	
7	138.50	0.3	0.688	0.167	0.22	
5	130.60	0.75	0.666	0.04	0.26	
16	27.35	0.35	0.659	0.0726	0.16	
11	89.30	0.25	0.656	0.143	0.36	
6	145.75	0.9	0.651	0.0825	0.43	
7	91.20	0.65	0.649	0.127	0.38	
10	34.75	0.25	0.648	0.204	0.32	
1	12.70	0.15	0.636	0.2	0.25	
14	65.65	1.3	0.634	0.0714	0.15	[1]
6	149.85	0.5	0.632	0.147	0.33	[70]
15	42.90	0.25	0.631	0.13	0.097	[64, 70]
4	52.35	0.35	0.631	0.086	0.26	
17	15.70	0.55	0.623	0.131	0.48	
1	241.85	0.25	0.62	0.0581	0.28	
16	18.65	0.2	0.616	0.103	0.47	
7	145.05	0.25	0.616	0.132	0.15	
4	110.45	0.25	0.616	0.312	0.23	
5	49.45	0.45	0.615	0.0395	0.4	
2	74.35	0.45	0.614	0.143	0.15	
17	40.90	0.85	0.614	0.159	0.22	[64, 117]
8	57.05	0.3	0.611	0.117	0.19	
11	91.95	0.25	0.61	0.092	0.48	
17	56.15	0.5	0.607	0.115	0.15	
11	71.10	0.4	0.607	0.102	0.068	
4	153.40	0.35	0.607	0.105	0.3	
22	20.10	0.25	0.605	0.152	0.18	[132]
2	130.60	0.25	0.605	0.174	0.5	
7	65.10	0.9	0.604	0.0656	0.33	
1	49.15	1.2	0.602	0.155	0.32	
20	25.05	0.6	0.601	0.0918	0.45	

Table 5.8: The first 35 estimated inversions from HapMap data, YRI data set, ordered by NBS score. The window size was 250 kb and each window had at least 20 SNPs. Estimated sample p-value was 0.00703.The data set had 200 haplotypes. The MAF column represents the proportion of the smaller inferred subset. See Section 3.4 for description of R_M. 'Chr.' is the chromosome of the inversion; 'Ref.' gives literature references.

Chr.	Start (Mbp)	Length (Mbp)	NBS	$\frac{R_M}{\#\text{SNPs}}$	MAF	Ref.
16	34.60	0.5	0.673	0.0952	0.23	
17	42.05	0.15	0.648	0.0357	0.41	
16	34.05	0.5	0.629	0.0536	0.23	[64]
5	49.45	0.4	0.608	0.0635	0.45	
7	56.35	0.3	0.604	0.14	0.48	
16	51.70	0.25	0.592	0.155	0.35	
7	65.10	0.95	0.589	0.0755	0.22	
16	28.70	0.15	0.586	0.188	0.2	
15	49.50	0.25	0.584	0.138	0.41	
6	26.60	0.3	0.575	0.11	0.42	[1, 64, 132]
9	94.00	0.5	0.568	0.0882	0.26	
19	55.15	0.25	0.567	0.174	0.26	[64]
5	138.05	0.25	0.565	0.132	0.34	
8	48.00	0.3	0.565	0.265	0.44	
15	81.00	0.1	0.562	0.2	0.42	
16	66.60	0.3	0.562	0.188	0.35	
12	81.20	0.25	0.561	0.213	0.27	
22	19.35	0.3	0.56	0.195	0.39	
8	99.90	0.75	0.559	0.104	0.29	[1]
10	64.55	0.45	0.559	0.153	0.28	
3	180.20	0.25	0.555	0.123	0.47	
17	19.95	0.4	0.554	0.208	0.39	
1	233.35	0.25	0.553	0.0976	0.4	
8	104.55	0.35	0.549	0.0676	0.28	
3	47.55	0.6	0.547	0.174	0.34	
18	32.60	0.5	0.547	0.117	0.38	
5	133.55	0.25	0.546	0.0926	0.33	
20	47.00	0.25	0.546	0.0959	0.44	
5	87.55	0.25	0.545	0.213	0.3	
7	143.45	0.25	0.544	0.125	0.41	[1, 64]
15	75.25	0.35	0.544	0.135	0.38	
4	103.90	0.25	0.544	0.161	0.29	
4	52.35	0.3	0.544	0.167	0.24	
2	148.25	0.3	0.543	0.0833	0.28	
7	99.75	0.2	0.543	0.14	0.3	

Table 5.9: The first 35 estimated inversions from HapMap data, JPT+CHB data set, ordered by NBS score. The window size was 250 kb and each window had at least 20 SNPs. The estimated sample p-value was 0.00012. The data set had 340 haplotypes. MAF column represents the proportion of the smaller inferred subset. See Section 3.4 for description of R_M. 'Chr.' is the chromosome of the inversion; 'Ref.' gives literature references.

Chr.	Start (Mbp)	Length (Mbp)	NBS	$\frac{R_M}{\#\text{SNPs}}$	MAF	Ref.
7	143.40	0.25	0.888	0	0.24	[1, 64]
1	25.50	0.25	0.805	0.13	0.27	[64]
2	110.15	0.2	0.769	0.156	0.4	[64]
20	29.25	0.45	0.766	0.137	0.28	
3	58.65	0.35	0.761	0.123	0.36	
17	22.35	0.3	0.744	0.167	0.45	
14	65.60	1.4	0.742	0.157	0.47	[1]
10	31.55	0.35	0.719	0.0714	0.21	
4	106.75	0.35	0.702	0.0769	0.11	
8	124.30	0.25	0.701	0.194	0.4	
6	90.40	0.25	0.699	0.103	0.29	
12	81.10	0.25	0.699	0.208	0.44	
11	3.25	0.3	0.697	0.32	0.47	
3	161.40	0.45	0.692	0.163	0.24	
1	191.15	0.55	0.69	0.152	0.48	
13	78.70	0.35	0.678	0.239	0.27	
5	102.20	0.5	0.671	0.129	0.46	
1	153.30	0.5	0.668	0.0842	0.26	
8	68.20	0.35	0.665	0.238	0.43	
1	108.55	0.25	0.661	0.0952	0.42	[64]
6	44.75	0.8	0.66	0.135	0.28	
7	110.55	0.35	0.651	0.409	0.5	
2	130.60	0.25	0.651	0.163	0.43	
16	22.20	0.25	0.649	0.08	0.14	[64, 70, 132]
4	52.35	0.45	0.647	0.162	0.37	
16	68.80	0.4	0.644	0.214	0.44	
3	196.80	0.25	0.644	0.222	0.26	[64, 132]
1	93.30	0.5	0.641	0.1	0.34	
8	81.75	0.25	0.64	0.136	0.42	
12	91.60	0.25	0.637	0.163	0.094	
17	58.30	0.6	0.636	0.203	0.21	
2	189.20	0.25	0.633	0.087	0.074	
5	70.70	0.3	0.633	0.167	0.38	
12	85.80	0.35	0.632	0.189	0.41	[64]
12	98.75	0.45	0.63	0.102	0.068	

For example, soon after the 900-kb inversion in chromosome 17 [117] is the region of chr17:42,050,000–42,200,000, which is reported in all three data sets, although in the JPT+CHB data set with additional 50 kb. This region does not correspond to an inversion listed in DGV. It is quite possible or even likely that there is another explanation for this region.

The references' fields in the tables are not very informative about the degree of overlap between the region reported in the literature and the one suggested by NBS-Scan or the true reason the regions were listed as candidates by NBS-Scan. For example, inversions that were reported by Ahn et al. [1] and intersected the candidate regions were short, less than 100 kb in length, making their reliable detection by NBS-Scan unlikely. In some cases, there were no SNPs within the known inversion region. Sometimes, the inversions were only partially covered by the candidate region.

Therefore, it is ill-advised to claim the reported MAF is an estimate of the proportion of the inversion arrangements, if the candidate region intersects a known inversion. In some cases, one candidate region contained multiple reported inversions, placing the connection between MAF and the proportion of a (single) inversion even more in question.

Before comparing the results of different inversion prediction algorithms, it is important to note that Bansal et al. [9] did their experiments on HapMap phase I data sets, Sindi and Raphael [112] originally on HapMap phase II data sets and we on HapMap phase III data sets. Furthermore, all three cases used different genome builds. The conversion of the predictions of [9] from one build to another has likely have been a disadvantage in the following comparisons. The results mentioned regarding Bansal et al. [9] in these investigations are what they report and not results of another implementation on phase III data sets. By comparison, the hybrid method, NBS-Scan and our implementation of SR-method used the same phase III data sets. Hence, interpreting the results of the comparisons of these algorithms should be done with care. The threshold for SR-method score for an inversion was set to 10^{-5} like Sindi and Raphael [112] did. This threshold value was used also by the hybrid method.

The inversion predictions of NBS-Scan are not similar to the predictions of SR-method. Table 5.10 contains the summary of

Table 5.10: Summary of NBS-Scan predictions intersecting with the predictions of SR-method.

	NBS-Scan predictions only	NBS-Scan-predictions intersecting SR-method predictions	SR-method predictions intersecting NBS-Scan-predictions	SR-method predictions only
CEU	501	5	24	69
YRI	121	2	8	74
JPT+CHB	600	10	23	34

the relationships between the predictions for each data set separately. Note that SR-method's predictions could overlap and such were counted as two distinct predictions. Overall, NBS-Scan gives a much higher number of candidate regions for the CEU and JPT+CHB data sets. If we considered only as many highest-ranking NBS-Scan predictions as SR-method's prediction list had, the number of NBS-Scan-predictions overlapping SR-method's predictions would be 4, 2 and 4 in CEU-, YRI- and JPT+CHB-data sets, respectively. These correspond to 4.3%, 2.4% and 7.0% of the number of predictions.

Of particular interest is that the 900-kb inversion in chromosome 17, which we first discussed on page 144, was not listed in Sindi and Raphael's [112] original candidate list for CEU data set. However, by using HapMap phase III CEU data set, it discovers an inversion candidate that is marked at chr17:41,377,578–42,217,772, whereas the position given in [4] is chr17:40,899,921–41,989,253, so the inversion is detected only partially.

NBS-Scan was designed particularly to detect such inversions, and it is more accurate about detecting this inversion. This very slightly suggests that NBS-Scan can possibly be considered to complement Sindi and Raphael's method.

As an elementary test of whether NBS can help SR-method in

detecting inversions, the predictions of SR-method and NBS-Scan were combined by taking their intersection. The performance of SR-method compared to the combined method was done by computing the proportion of known inversion polymorphism regions listed in DGV [59] on the July 2009 to the predicted regions.

In Table 5.11 we see that with the exception of the YRI population, the fraction of basepairs known to be inside inversion polymorphisms out of all predictions decreases. The intersection operation also eliminates at least 80% of the basepairs in SR-method's predictions.

The table also contains as comparison the predictions of Bansal et al. [9], which have been lifted over to NCBI build 36. The predictions of NBS-Scan by itself seems to be more accurate than those in Bansal et al. [9], but on the other hand, some of their predictions were lost in the lift-over process. They also used a different data set, which also might cause their predictions to be less accurate.

Additionally, the table contains the values computed by using the hybrid method described in Section 3.5. The entropy distributions and the empirical likelihood ratio distributions were computed separately for each of the data sets. The main differences to the experiment setup of Sindi and Raphael [112], beside the differences to the algorithm as described in Section 3.5, are in that the SNPs had to have MAF of at least 0.05 instead of 0.1, and that the data sets used were HapMap phase III data sets, with the note that CEU and YRI data sets used only the subset of haplotypes that were phased only from trios. Note that the upper limit of the degrees of freedom in deciding whether a pair of potential inversion end-points was to be considered was unchanged in spite of the higher number of haplotypes in the data sets. The hybrid method approached the problem of clustering nearly identical candidates so that a prediction was removed from consideration if there was another prediction with smaller p-value and both endpoints within 10 kb of the respective endpoints of that prediction.

If we approximate the human genome to be 3 billion basepairs long, then the inversions listed in the database cover 1.75% of the whole genome. This also means that the trivial prediction algorithm that labels every basepair as inverted gets the precision of 1.75%; this is useful in interpreting the values in Table 5.11.

Table 5.11 also shows that the hybrid method, which is essen-

Table 5.11: Comparison of different inversion predictions. The values are the fractions of known true positive predictions out of all positive predictions in terms of single basepairs, based on DGV entries on July 2009. The intersection of SR-predictions with NBS-Scan means those basepairs that are predicted to belong in inversions by both methods. Note that the Bansal et al. [9] used HapMap phase I data sets whereas our SR-method implementation, NBS-Scan and the hybrid method used HapMap phase III data sets.

	CEU	YRI	JPT+CHB
Bansal et al. [9]	4.8%	2.5%	1.9%
Sindi and Raphael [112] on HapMap phase III data	37.8%	16.8%	21.5%
NBS-Scan	2.2%	3.8%	2.8%
Hybrid method	36.7%	11.8%	21.5%
SR-predictions on HapMap phase III data intersected with NBS-Scan	25.5%	44.4%	14.0%

tially a slightly tweaked version of Sindi and Raphael's method, performs overall notably worse than the original. Let us investigate the differences between the methods further.

Comparisons between SR-method, NBS-Scan and the hybrid method were done on the same HapMap phase III data sets. Figure 5.12 displays the precision-recall curves for these methods for three different HapMap phase III populations. The basepairs covered by inversions in Database of Genomic Variants [59] were considered the set of positives. In precision-recall curves, 'precision' is the number of true positives divided by the number of all predicted positives. The other axis, 'recall', is the number of true positives divided by the number of all positives, i.e., how much of all inversion polymorphisms in the database are covered by the list of predictions. By including one candidate region at a time to the set of candidates, we can compute each node on the curve. A direct vertical drop means the prediction that was added did not coincide with a known inversion.

For NBS-Scan, the parameters used to generate the list of candidates was fixed. These lists of candidates were sorted by their

NBS score and the precision-recall curve was drawn from this list of candidates. This means that the curves do not look identical with different parameters used to generate the lists.

In Figure 5.12 we see that NBS-Scan is at its best at very short candidate lists (low recall). In particular in the YRI population, NBS-Scan and the hybrid method perform relatively better than SR-method when only a few candidates are given.

The curves are plotted based on the estimated predictions. Note that the candidate lists with NBS-Scan have notably more items than those of the SR-method and the hybrid method, but the recall rate is in two cases less.

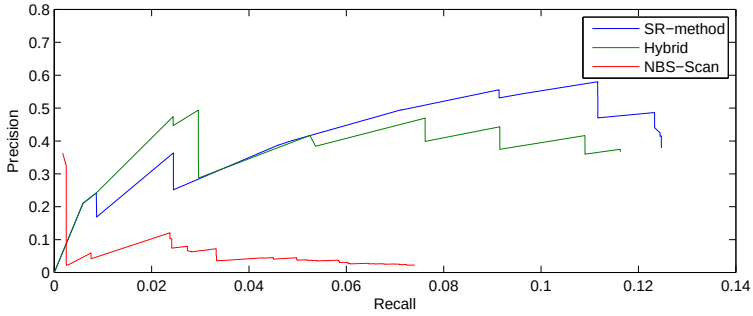
Figure 5.12 shows a notable decrease in the difference in the performance of NBS-Scan and the hybrid method in comparison to SR-method.

There are three apparent potential reasons for this beside the better performance of NBS-Scan and hybrid method. First, the first predictions of SR-method are actually inversions, but they are not listed in DGV, for instance, because they are not known. Second, the data set has some underlying property that makes it difficult for SR-method to perform on. Third, the low precision is due to random variation. The reasons for this are a potential topic for future studies to investigate how past population histories affect the accuracy of different inversion detection algorithms.

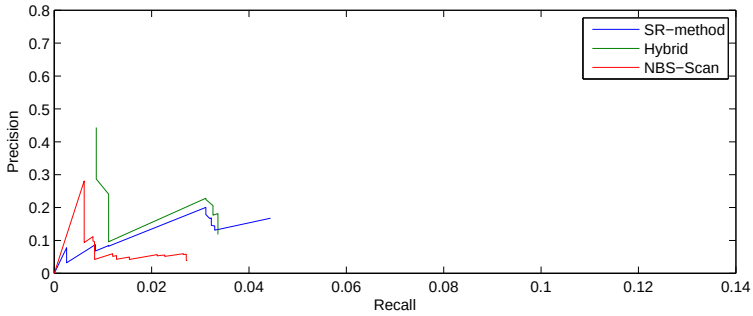
Inversions characterized by Antonacci et al.

Antonacci et al. [4] investigated six inversions in three different HapMap data sets: CEU, YRI and JPT+CHB. In total, they listed estimated inversion haplotype frequencies based on 54 sampled haplotypes across the three data sets. These inversions are listed in Table 5.12.

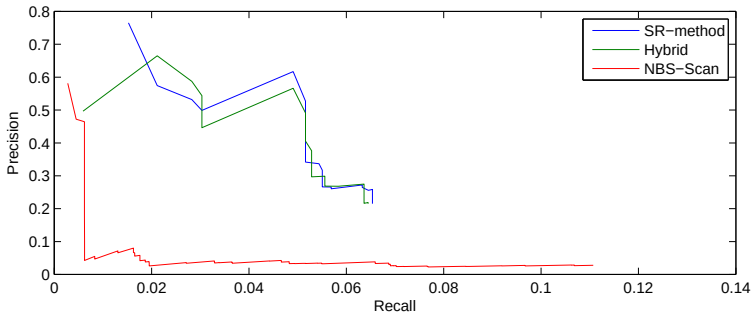
Four of the six inversions were listed in the candidate regions produced by NBS-Scan (Tables 5.7–5.9) at least partially: the inversions in chromosomes 15 and 17. Antonacci et al. mention that based on SNP haplotypes, they could accurately predict the inversion status in the inversions of chromosomes 3 and 8, and also the 900-kb inversion in chromosome 17. Also, Deng et al. [28] used principal component analysis (PCA) to correctly deduce the haplotype orientations from SNP data in 418 haplotypes from the HapMap data for the chromosome 8 inversion. This finding is in contrast to



(a) CEU data set



(b) YRI data set



(c) JPT+CHB data set

Figure 5.12: Precision-recall plots of inversion detection. The precision and recall were computed based on predicted basepairs instead of predicted segments with known inversions in Database of Genomic Variants [59] in the July 2009 NCBI build 36 release acting as the set of positives. All methods used the same HapMap phase III data sets.

Table 5.12: Statistics of six inversions investigated from Antonacci et al. [4]. The coordinates were lifted to NCBI Build 36.1 coordinates with the liftOver tool (http://genome.ucsc.edu/cgi-bin/hgLiftOver).

Inversion mapping	Length	Highest frequency (Population)	Total frequency
chr3:196,882,966–198,870,687	1.9 Mb	12.5% (CEU)	7.4%
chr8:7,225,962–12,487,029	4.7 Mb	59.1% (YRI)	42.6%
chr15:28,524,207–30,602,466	2 Mb	25.0% (CEU)	20.4%
chr15:72,151,413 73,356,183	1.2 Mb	6.2% (JPT+CHB)	1.9%
chr17:31,888,441–33,393,152	1.5 Mb	9.1% (YRI)	5.5%
chr17:40,899,921–41,989,253	900 kb	18.7% (CEU)	5.5%

the inversions NBS found, which did not include the inversions in [4, 28] for chromosomes 3 and 8.

It is of interest that the inversion at chr17:31,888,441–33,393,152 was not intersected by the inversion candidates in the YRI population, in which Antonacci et al. estimate the inversion to be present at the highest frequency. Instead, the candidate lists for the two other data set intersect with the inversion.

The inversions chr8:7,225,962–12,487,029 and chr15:28,524,207–30,602,466 (Figures 5.13 and 5.14, respectively) were chosen here as examples. The haplotypes have been sorted by spectral ordering, but the inferred data set subdivision is not displayed for the inversion in chromosome 8, as NBS could not detect it.

Both inversions have many SNPs, but they nearly completely lack division into two distinct groups detectable by NBS for some reason. Bosch et al. [13] discovered that SNPs near the ends of the inversion in chromosome 8 could be used to infer the inversion status.

It is interesting to note that NBS slightly peaks in the latter case near the inversion end-points. It is possible that this is due to the low number of SNPs in those regions, which might increase

NBS due to chance. An alternative explanation would be that the inversion end-points and their immediate neighbourhood have retained the division in two while gene flow over generations has removed the apparent distinction between the two arrangements. SNPs with positions in ranges 28.5–28.8 Mb and 30.4–30.7 Mb, i.e., SNPs near the inversion breakpoints, had NBS score of 0.63, slightly supporting the latter explanation. However, the data had only 12 SNPs within those regions and only two of them in the latter range, making the value of this investigation low. The possibility of using clustering similarity measures to investigate how similar the inferred bipartitions at both ends separately are is ill-advised due to the low number of SNPs in one of the regions.

It should also be noted that these inversions highlight what is considered a correct detection of an inversion in this thesis: the candidate region intersects at least partially with a known inversion. In this case, only a 300 kb subsegment of the 2 Mb inversion in chromosome 15 was detected. Because most of the inversion does not seem to be arranged in the four-field pattern, this small segment might not have been included in the candidate set because of the inversion but some other cause to form such a haplotype structure.

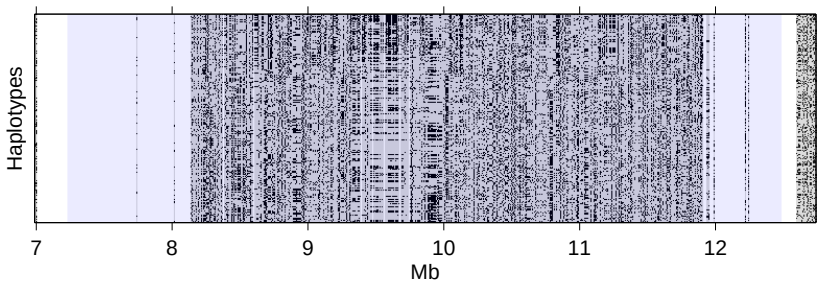
Detected known inversions

Two candidate regions that contained previously validated inversions were selected for further examination. The first case, a short inversion in chromosome 22, is an example of the actual inversion region containing no genotyped SNPs with MAF ≥ 0.05 (Figure 5.15). It is questionable whether the bipartition visible at 20.25–20.35 Mb is due to the presence of the inversion or if it is by chance.

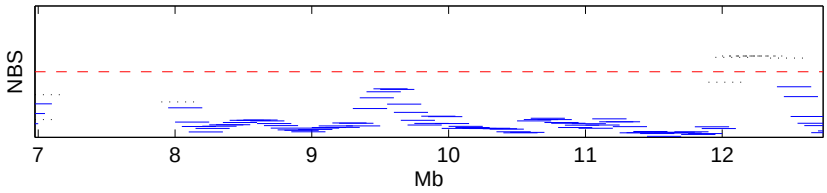
Another case is the a longer inversion in chromosome 16 (Figure 5.16). The bipartition is visible also before the inversion. The pattern is broken soon after the inversion, though, but the region still displays lowered recombination rate. Within the inversion, the four-field pattern seems to hold well, but not perfectly.

Candidate regions

Based on the visible four-field pattern signal, two candidate regions proposed by NBS-Scan were selected for further examina-



(a) SNP data. Bin size is 7681 bp. The inversion region is highlighted in blue. One SNP is chosen from each bin to represent it. The haplotypes have been sorted by spectral ordering as described in Section 3.3.



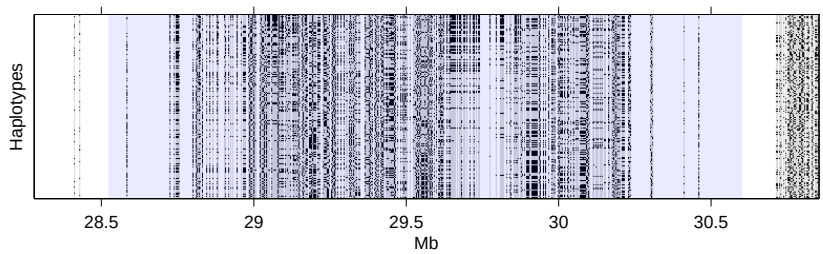
(b) The NBS score in windows of 250 kb. The scale is from 0 to 1, the red dashed line representing the detection threshold value of 0.5. Windows with less than 20 SNPs are marked by black dotted lines.

Figure 5.13: Data view of an inversion chr8:7,225,962–12,487,029 in YRI data set after spectral ordering. The shown window includes the inversion and 250 kb of context in both directions.

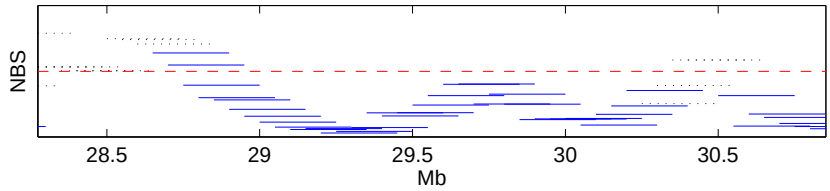
tion: chr6:44,750,000–45,750,000 in the JPT+CHB data set and chr2:74,350,000–74,800,000 in the CEU data set. The SNP and NBS plots are Figures 5.17 and 5.18.

Neither of these two ranked very high in the lists of candidate regions. In both cases, however, the region of high NBS was sharply defined, as the bipartition vanishes nearly instantly outside the candidate regions. These regions have nonetheless an interesting recombinational history, even if the reason for that might not be an inversion.

In both cases, NCBI Map Viewer for Build 36.3 shows several genes within the suggested regions especially in the case of the chromosome 2 candidate. This decreases the probability of these four-field patterns having been formed due to inversions.



(a) SNP data. Bin size is 3437 bp. The inversion region is highlighted in blue. One SNP is chosen from each bin to represent it. The haplotypes have been sorted by spectral ordering as described in Section 3.3.



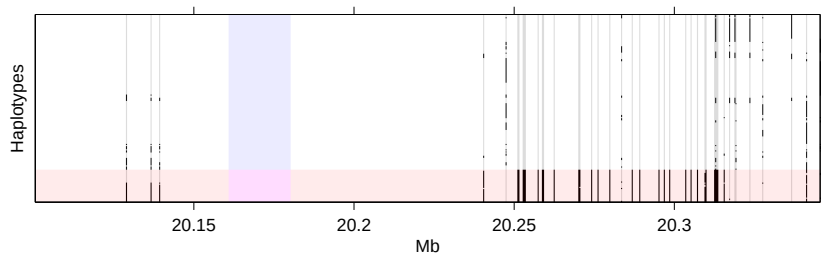
(b) The NBS score in windows of 250 kb. The scale is from 0 to 1, the red dashed line representing the detection threshold value of 0.5. Windows with less than 20 SNPs are marked by black dotted lines.

Figure 5.14: Data view of an inversion chr15:28,524,207–30,602,466 in CEU data set after spectral ordering. The shown window includes the inversion and 250 kb of context in both directions.

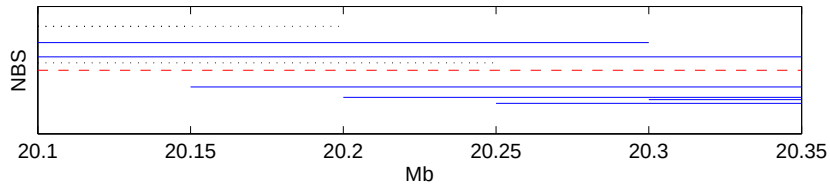
Comparison of NBS histograms

The distribution for NBS varies considerably in different data sets. To study this difference, the NBS distributions in several different data sets were computed. First, the distributions computed from HapMap data sets with windows containing at least 20 SNPs were used to generate real-life NBS histograms. Second, InvCoal and COSI were used to produce 3,000 data sets without inversions using both 48-haplotype panel ascertainment; for InvCoal, also Carlson's tag-SNP algorithm was tried. The resulting frequency histograms are displayed in Figure 5.19. The purpose of this was to investigate how accurately the simulator data can reproduce the NBS distribution seen in real data sets. Inversions cover only a small part of the human genome, so their effect on the HapMap distributions is minor except for the tail.

The histograms clearly indicate the InvCoal simulator is not able



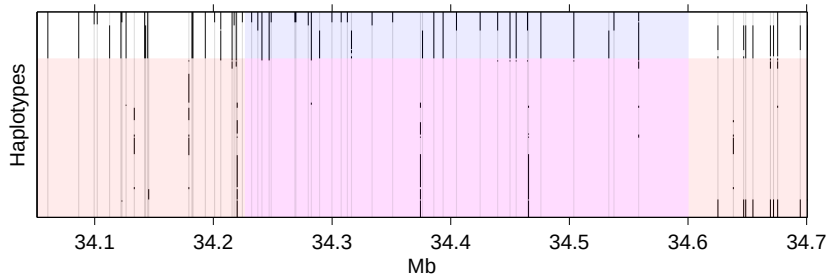
(a) SNP data. Bin size is 333 bp. One SNP is chosen from each bin to represent it. The haplotypes have been sorted by spectral ordering as described in Section 3.3. The division that maximizes the NBS score is marked by the red/white background.



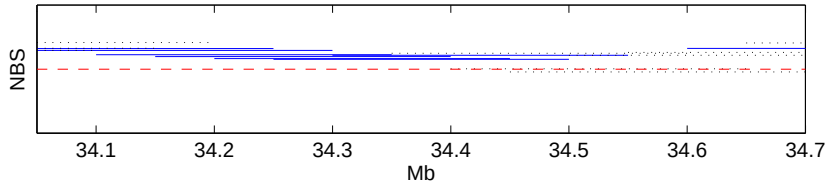
(b) The NBS score in windows of 250 kb. The scale is from 0 to 1, the red dashed line representing the detection threshold value of 0.5. Windows with less than 20 SNPs are marked by black dotted lines.

Figure 5.15: Data view of chr22:20,100,000–20,350,000 in the CEU data set. The known inversion (20,160,891–20,180,436) [132] is highlighted in blue. The spectral ordering in (a) is computed based on the whole viewed segment. The division that maximizes NBS for the viewed segment is also displayed.

to produce data that would behave like the human genome on average with the parameters used in the experiments. Therefore it is ill-advised to consider the results from synthetic experiments as indicative of the performance of NBS on human chromosomes. But although NBS histograms for InvCoal-simulated data do not match any of the HapMap NBS histograms, it might be difficult to obtain a better matching of these histograms with the limited population and recombination model that InvCoal has. By comparison, COSI simulations fit the real-life data well in all three subpopulations, suggesting that the coalescent can be used to generate realistic data also in this respect. It is likely that the difference in the histograms between COSI and InvCoal is in part due to modelling the subpopulation and chiasma position distributions differently. This casts



(a) SNP data. Bin size is 867 bp. One SNP is chosen from each bin to represent it. The haplotypes have been sorted by spectral ordering as described in Section 3.3. The division that maximizes the NBS score is marked by the red/white background.

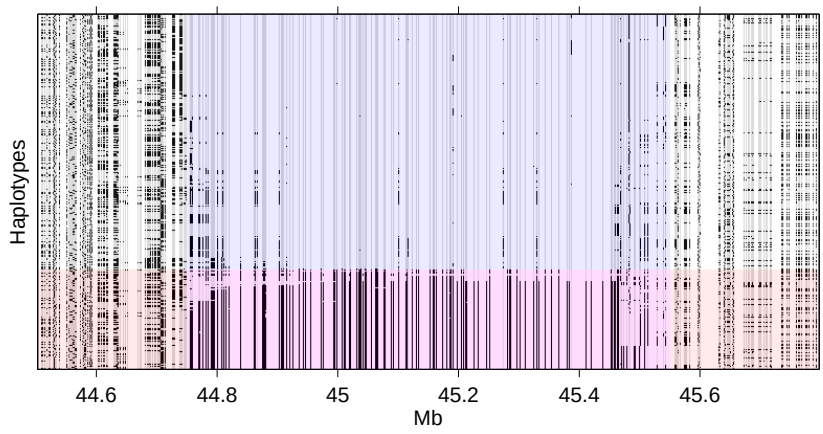


(b) The NBS score in windows of 250 kb. The scale is from 0 to 1, the red dashed line representing the detection threshold value of 0.5. Windows with less than 20 SNPs are marked by black dotted lines.

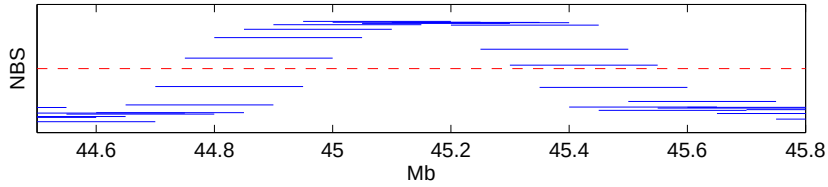
Figure 5.16: Data view of chr16:34,050,000–34,700,000 in the YRI data set. Note that the region proposed by NBS-Scan was chr16:34,050,000–34,550,000. The known inversion (34,226,853–34,599,997) [64] is highlighted in blue. The spectral ordering in (a) is computed based on the whole viewed segment. The division that maximizes NBS for the viewed segment is also displayed.

notable doubt on the performance of NBS in practice.

The difference between YRI and the other HapMap data sets seen in Figure 5.19 is interesting. Because the SNPs have undergone the same ascertainment process, the detected difference corresponds to a difference in the actual populations. Although the best-fit parameters of COSI [104] should not be considered as a statement of the population past (the authors even warn against doing so), the effective population size parameter of the African subpopulation is notably higher than the European and Asian subpopulations. On the other hand, a similar statement has been made by Tenesa et al. [126] with a higher estimate of effective population size for the



(a) SNP data. Bin size is 2000 bp. One SNP is chosen from each bin to represent it. The haplotypes have been sorted by spectral ordering as described in Section 3.3. The division that maximizes the NBS score is marked by the red/white background.



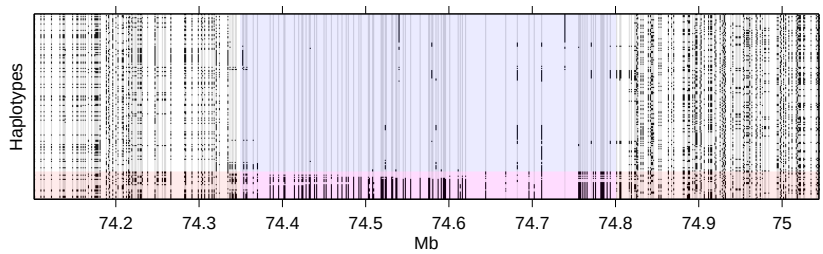
(b) The NBS score in windows of 250 kb. The scale is from 0 to 1, the red dashed line representing the detection threshold value of 0.5.

Figure 5.17: Data view of a putative inversion chr6:44,750,000–45,550,000 (highlighted in blue) in the joint JPT+CHB data set.

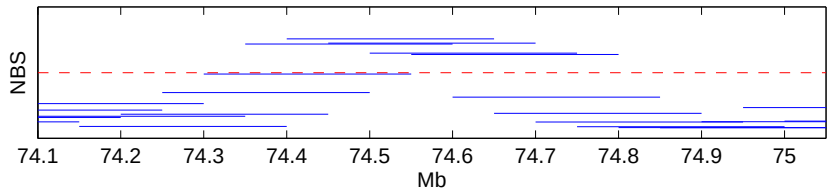
population of the YRI data set in comparison to those of CEU, JPT and CHB data sets.

The NBS histogram displays why the YRI data set resulted in the shortest candidate list of the three populations as it has the smallest proportion of NBS scores over the detection threshold of 0.5 but has a large number of windows with very low NBS scores. However, the estimated p-values for the result sets of YRI data set were not the best of the three subpopulations.

The precision-recall curves (Figure 5.12) showed that NBS performed best in comparison with other methods on the YRI data set with only a short candidate list (low recall). This is in accordance with the belief that NBS performs better with populations where



(a) SNP data. Bin size is 1266 bp. One SNP is chosen from each bin to represent it. The haplotypes have been sorted by spectral ordering as described in Section 3.3. The division that maximizes the NBS score is marked by the red/white background.



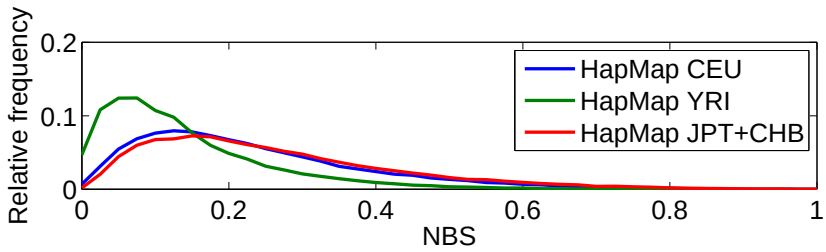
(b) The NBS score in windows of 250 kb. The scale is from 0 to 1, the red dashed line representing the detection threshold value of 0.5.

Figure 5.18: Data view of a putative inversion chr2:74,350,000–74,800,000 (highlighted in blue) in the CEU data set.

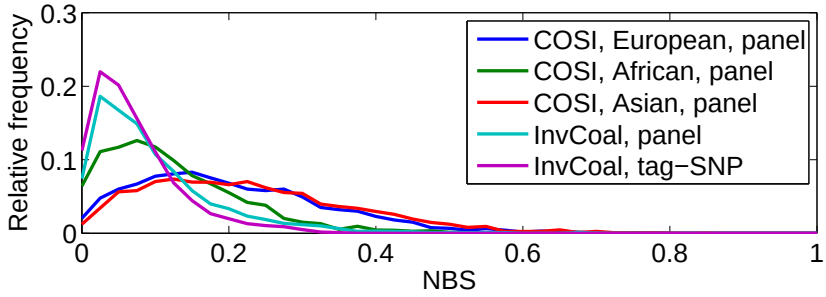
the haplotype blocks are shortest.

Since the simulated data had no inversions, the tails of NBS values of COSI simulations can be used to estimate the false positive rate of NBS in real data, if we assume that the simulated ascertainment process does not skew the NBS histogram much and the NBS scores of nonoverlapping windows are independent even in the same chromosome. For the simulated European data sets, 94 windows of 3,000 replicates have NBS greater than 0.5, giving a non-inverted window the probability of 0.0313 of being falsely claimed as a potential inversion. For the simulated African and Asian data sets, the probabilities were 0.0060 and 0.0480.

If we count the number of non-overlapping 250 kb windows with at least 20 SNPs in them for each HapMap data set and multiply the number with the respective probability of calling a non-inverted window as inverted, which we estimated above, we get an estimate of how many 250 kb windows are mistakenly called inverted by



(a) Distribution of NBS in HapMap data sets



(b) Distribution of NBS in synthetic data sets. InvCoal used the same parameters as in the synthetic experiments section; all curves are plotted based on 3000 250-kb-long data sets with 120 haplotypes.

Figure 5.19: NBS distributions computed from different data sets with 250-kb windows.

NBS-Scan. These numbers are 329, 63 and 502 for CEU, YRI and JPT+CHB data sets, respectively. If each of these is considered an independent false candidate region, this leaves us with 177, 60 and 108 candidate regions in the respective data sets that might be real inversions. These correspond to 35%, 49% and 18% of all candidate regions. It should be noted that these numbers are only slightly educated guesses, as the simulation process and the properties of the simulated data differ from those of reality and the real data sets. Yet, assuming the number of false positives is correct, the candidate set after removing those candidates intersecting known inversions would still contain more regions than the estimated number of false positives. Therefore it is possible that NBS-Scan finds useful information of real previously unvalidated inversions.

5.2.5 Discussion

It appears that not many frequent inversions can be identified by using NBS. To some extent, the lack of a detectable four-field pattern in most of the inversions characterized by Antonacci et al. [4] casts doubt also on the accuracy of the output of InvCoal with the chosen parameters.

However, based on the computed p-values for the results, it appears that NBS can, with properly selected parameters, perform better than a random guess also on real data. The exact reason for this, however, is unclear, as in some cases the known inversions are only small parts of the suggested inversion region and may not contain any genotyped SNPs.

In their paper on copy number variant (CNV) detection, McCarroll et al. [82] also discussed the different size estimates of CNVs by using different detection methods. In particular, some methods resulted in much larger estimates for the same variants. If this holds true also for inversion polymorphisms in DGV, it may affect also the estimated p-values for the result set.

Phasing errors (errors in assigning which allele belongs to which of the two copies of a chromosome) may also affect the performance of NBS. Such errors can considerably affect the performance of NBS, as their effect spans several SNPs. Because of this, the candidate regions of JPT+CHB data set, which has only unrelated individuals for which phasing is harder, are possibly less reliable than those of CEU and YRI data sets, which have in our case only trios, for which phasing is easier. It should be noted that phasing errors were not a factor accounted for in estimating the number of correctly detected inversions.

The experimental results gained by using InvCoal to generate data sets should be carefully considered, as the score distribution under null hypothesis is notably different from the real data sets. Yet, the effects of different simulation parameters can be expected to be generally valid: The more frequent the inversion is, the more likely it is to be detected. Increasing the number of haplotypes does not improve the accuracy of NBS, but the recombination rate has a significant effect on the power of NBS.

Another relevant factor to consider is the age of the inversion. The simulation experiments showed somewhat surprisingly that old

inversions would be more likely to be detected than young ones. Old inversions, however, can be expected to have either become fixed or extinct in the population, and as such, may be expected to be present in fewer numbers than younger inversions. This would effectively decrease the probability of NBS-Scan detecting a randomly chosen inversion in the (human) genome.

Most of the experiments, including the comparison experiment with Sindi and Raphael's method [112], did not use any kind of recombination measure to filter out false positives. The results on real data seemed not to prefer the filtering by R_M, judging by the p-values resulting from different minimum threshold values for detected recombinations per SNP. By comparison, in the simulation experiments R_M seemed to increase the number of true positives for low numbers of false positives.

This suggests that if properly done, using recombination measures to remove regions of otherwise low recombination rate may help in improving the performance of NBS-Scan and other similar methods to lower the false positive rate. Unfortunately, it is not known which is the best way to accomplish this.

The difference in the performance of NBS between real data and synthetic data is notable. This casts noticeable doubt on how realistically InvCoal can actually simulate data. As such, the results of the experiments on the synthetic data should be taken with a grain of salt.

5.3 Deletions

In this section, Kohler and Cutler's microdel [68] is compared to Deldec-Scan, the method based on haplotype frequency estimation and described in Chapter 4.

As an EM-based algorithm, Deldec was specified the maximum number of iterations for each restart in each window (this was set to 200) and the number of restarts (this was set to 5). The stopping criterion for one restart was either reaching the maximum number of iterations or the relative increase in the log-likelihood being below 10^{-7}.

The experiments were executed in parallel on servers with eight Intel Xeon 2833 MHz processors and 32 GB of RAM, running

Table 5.13: Parameters used to generate synthetic trio and unrelated data sets with Kohler and Cutler's simulator.

Parameter name	Parameter values
Number of trios	30, 100, 500, 1000
Deletion length	20 kb
Deletion frequency	0.5%, 1%, 5%, 10%, 20%

Ubuntu Linux 8.10.

5.3.1 Generating synthetic data

To generate synthetic data for detecting deletions, the simulator used by Kohler and Cutler[2], which they used in testing their deletion-detection program microdel [68], was used. The parameters used were the same as in their work, with the exception of using the simulator to produce 1,000 haplotypes for each simulated data set from which the trio haplotypes were sampled, limiting to each deletion being 20 kb in length and having mean SNP spacing of 2 kb. To summarize, Kohler and Cutler estimated the means and variances of miscall and no call rates per SNP for 8 different genotyping centres used in the HapMap project. These are then used to parametrize Beta distributions from which the error rates are sampled for each SNP independently by the simulator. Each simulated segment was 250 kb long. The author gratefully acknowledges the help of Assistant Professor David Cutler with the simulator.

For this thesis, 100 data sets with each of these estimated error rate parameters were generated, totalling 800 data sets for each parameter configuration. The rest of the parameters were left unchanged. For a list of the parameters and the values they were given, see Table 5.13.

For the case of unrelated data, the same simulator and parameters were used but the child genotypes were discarded from the data. However, the number of genotyped individuals in both trios

[2] `http://cutler.igm.jhmi.edu/Software/software.html` (Accessed 02.11.2009)

and unrelated individuals were kept the same by changing the number of simulated trios, thus making the results comparable.

The used SNP ascertainment scheme was built into the simulator; the scheme [75] considered all SNPs as independent in the ascertainment and accepted the SNP with probability dependent on the allele frequencies alone; this probability was modelled to correspond to the probability of such a SNP being in the SNP database dbSNP [111]. The used coverage parameter, η, was set to 7. See [75] for the meaning of η.

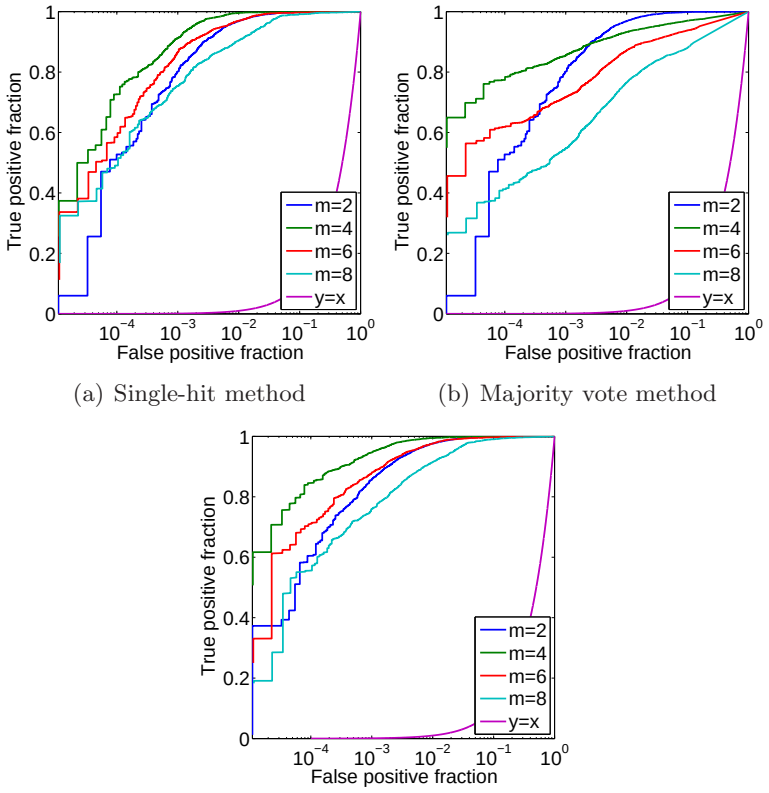
5.3.2 The power of deletion detection

The increase in the window size m beyond 4 SNPs decreased the per-SNP detection accuracy. Even though the signal carried by the SNPs within the window, presuming it is fully contained inside the deletion, is detected more reliably, the number of windows that include the deletion end-points is also higher and this makes the accurate detection of the deletion status of SNPs near the deletion ends more difficult. This can be seen in Figure 5.20 and Figure 5.21, where the change in the simulated deletion frequency results in the larger window no longer being the best for SNP-wise detection.

It appears that windows with size near 4 is in many cases a good choice (Figure 5.20) while using the mean method for SNP-wise assessment at least for deletion frequency 0.1.

Let us start with a comparison of running times. Of the tested algorithms, microdel was the fastest by a clear margin in larger data sets (Figure 5.22). Somewhat surprisingly, Deldec for unrelated individuals performed slower than for trios with as many genotyped individuals. Even though one iteration of the EM-algorithm per one individual includes in the case of trio data two calls of Yates' algorithm and in the case of unrelated individuals one call per individual, the number of iterations the EM-algorithm differs greatly between the cases.

The measured running times for Deldec-Scan include the time spent in estimating parameters the miscall and no call rate parameters τ and δ only in the case of trio data. The time spent in the haplotype frequency estimation EM-algorithm was measured as only the CPU time spent in user mode, as the implementation that scanned over the SNP data set for Deldec-Scan was very IO-heavy.

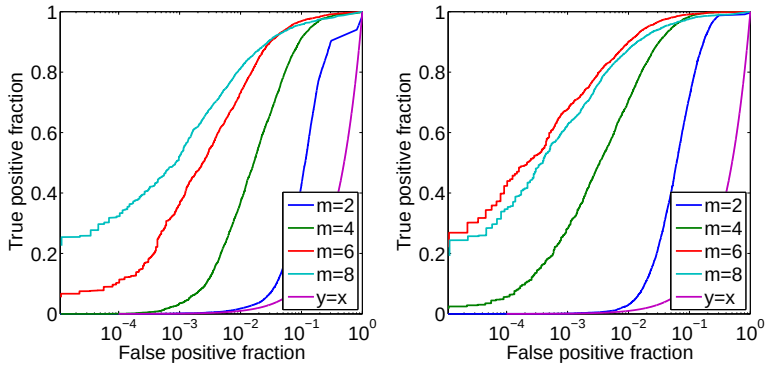


(a) Single-hit method

(b) Majority vote method

(c) Mean method

Figure 5.20: Deldec-Scan ROC curve comparison of accuracy of deletion end-point estimation for different window sizes m. There were 100 trios in each data set and deletion frequency f_0 was 0.1. The fractions are computed from SNP-wise deletion-status predictions.



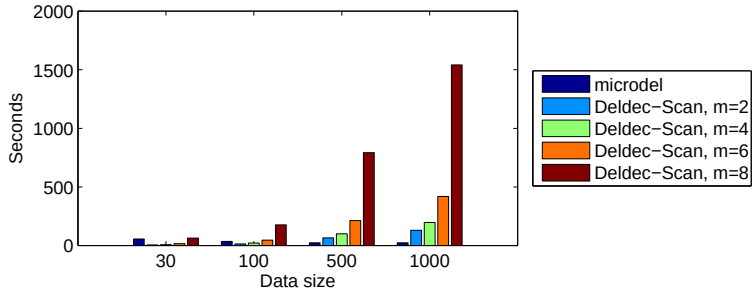
(a) $f_0 = 0.005$ (b) $f_0 = 0.01$

Figure 5.21: Deldec-Scan ROC curve comparison of accuracy for different window sizes m and deletion frequencies f_0 with the mean method. The data had 1,000 trios. The fractions are computed from SNP-wise deletion-status predictions.

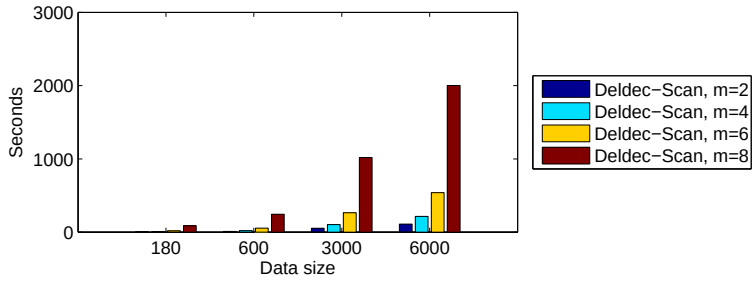
With a good implementation, the time spent to read the whole data set into the memory at once would likely not have resulted in a large time spent handling input-output calls. The measured times for microdel also contained only time spent in user mode. The times are averaged over different proportions of deletions.

Let us next consider deletion detection performance. In most of the cases, microdel outperformed Deldec-Scan (Figure 5.23). This might in part be explained by the simulator used to generate the synthetic data using the same error model as microdel and Deldec failing to estimate the error rates reliably for the trio data. Figure 5.24 displays the estimated genotyping error τ (see Section 4.2.2) histogram for the synthetic data sets from one genotyping centre. The mean genotyping error for the particular centre is denoted by in the figure by $\bar{\tau}$. Table 5.14 lists the false positive fractions for microdel, which were used also for deciding the points on Deldec-Scan's ROC curves by which to report the true positive fraction in Figure 5.23.

Surprisingly, it appears that the estimate is biased downwards. In the cases with $f_0 = 0.2$, the estimated genotyping error rate should have been constantly higher than the true genotyping error rate, because the estimation was done under the assumption of no



(a) Trios



(b) Unrelated individuals

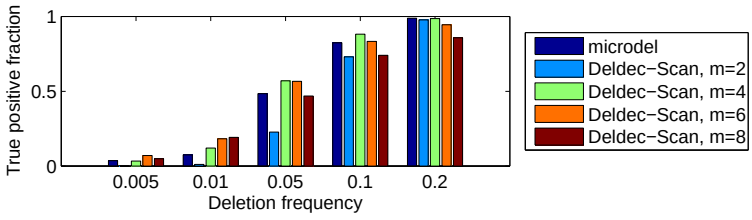
Figure 5.22: Average running times for different deletion detection algorithms under varying scenarios.

deletion being present. This is possibly due to the differences in the error models used in the simulation and in the estimation.

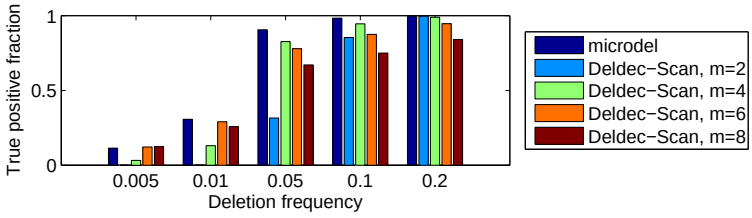
Deldec-Scan performed better than microdel only on small data sets (Figure 5.23(a)). This might also be due to microdel having a considerably higher false positive rate in the case of 30 trios, although this is in disagreement with the case of $f_0 = 0.05$ and 100 trio data sets (center set of columns in Figure 5.23(b)).

In the case of unrelated data, the false positive fraction was fixed at 0.0001 because microdel could not provide a baseline for the false positive fraction as it could not be used on data sets of unrelated individuals. The performance of Deldec-Scan is shown in Figure 5.25. In these simulations the error parameters discussed in Section 4.2.2 were set to $\tau = 0.001$ and $\delta = 0.01$. These values are reasonably close to the means of the errors of the genotyping centres estimated by Kohler and Cutler.

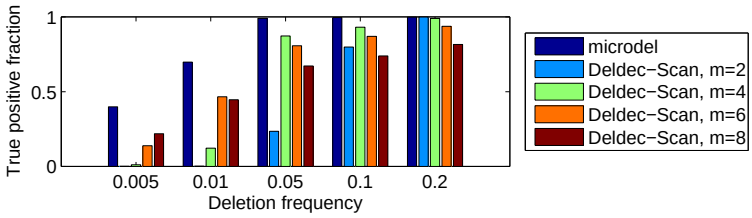
Surprisingly, with the largest data sets and deletion frequency



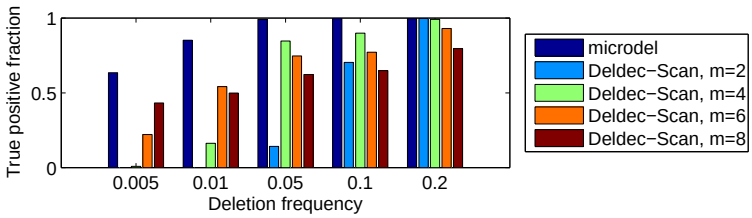
(a) 30 trios



(b) 100 trios



(c) 500 trios



(d) 1000 trios

Figure 5.23: True positive fractions for different deletion detection algorithms under varying simulated scenarios. The false positive fraction was fixed for Deldec-Scan to the one given by microdel.

Table 5.14: False positive fractions for microdel that were used also for Deldec-Scan in Figure 5.23.

	f_0				
Trios	0.005	0.01	0.05	0.1	0.2
30	0.0001	0.0009	0.0013	0.0022	0.0021
100	0.0001	0.0004	0.0012	0.0009	0.0006
500	0.0003	0.0004	0.0003	0.0002	0.0001
1000	0.0004	0.0004	0.0002	0.0001	0.0001

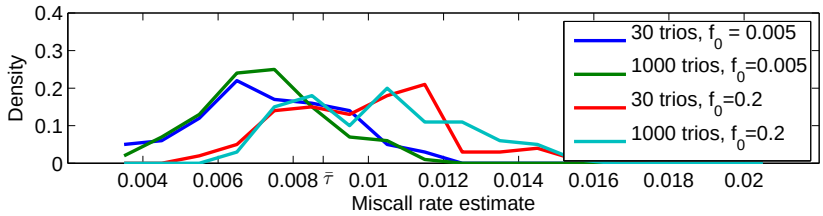


Figure 5.24: Genotyping error estimates under four different scenarios from one simulated genotyping centre, gained by the EM-algorithm described in Section 4.4. The tick $\bar{\tau}$ marks the mean genotyping error in the model of Kohler and Cutler [68].

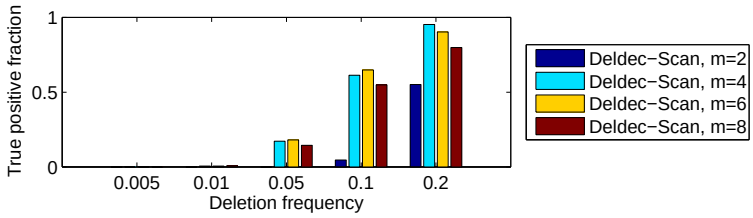
of 0.05 it appears that using unrelated individuals results in better power than with trio data sets with as many genotyped individuals. For example, the case of 30 trios is compared to the case of 90 unrelated individuals. However, because the power of the tests in data sets of unrelated individuals remains very low for rare deletions before a sudden rise, using trio data would still be preferable.

When comparing the case of data from unrelated individuals to trios, the ROC curves look different. In Figure 5.26(a-c) it can be seen that with the exception of window size $m = 2$ the rate of false positives is much higher than in comparable positions with trios (Figure 5.20(c) and Figure 5.21(b)). As Figure 5.25 show, with low deletion frequencies the power is much lower. Finally, the mean method seems to perform the best also for this kind of data (Figure 5.26(d)).

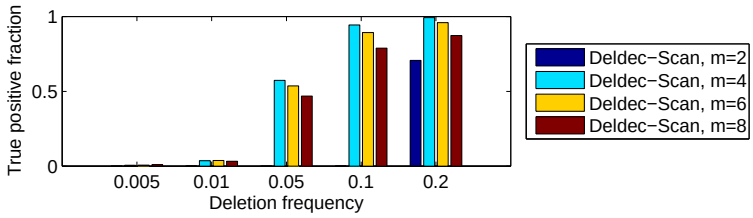
To measure the false positive fraction of Deldec-Scan with realistic SNP data sets without deletion, we used COSI [104] to generate 500 SNP data sets of 500 kb in length with 30 trios and mean SNP spacing of 2 kb. This corresponds to 250 Mb of simulated segments and 125,000,000 potentially deleted SNPs. The haplotypes were sampled from the simulated European population and underwent similar ascertainment simulation as in Kohler and Cutler's simulator with coverage parameter $\eta = 7$, after which they were randomly thinned so that each data had 250 SNPs. The errors were generated by following the error model of Deldec with parameters $\tau = 0.001$ and $\delta = 0.01$. For unrelated data otherwise similar data sets but with 90 individuals were simulated. The mean method was used to assign deletion status to SNPs. If we considered the mean of the log-likelihood ratios computed for the windows that each SNP was in to be useable in a likelihood ratio test using the χ^2 approximation and required the test statistic to be larger than the threshold corresponding to a p-value of 10^{-10}, the number of SNPs predicted to be deleted was 0 for both unrelated and trio data with $m \in \{2, 4, 6, 8\}$. With such an extreme threshold, the false positive fraction in real experiments could be low also in real data sets.

5.3.3 Real data

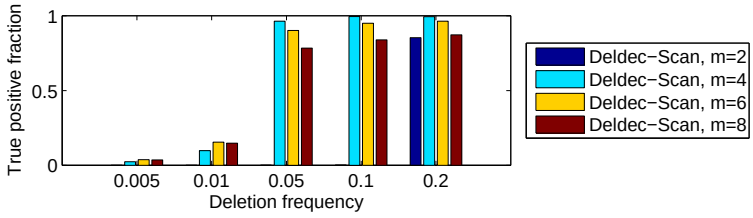
For detecting deletions, the data used are unfiltered HapMap Phase II data sets from January 2007 (rel. 21a) [129]. The coordinates of



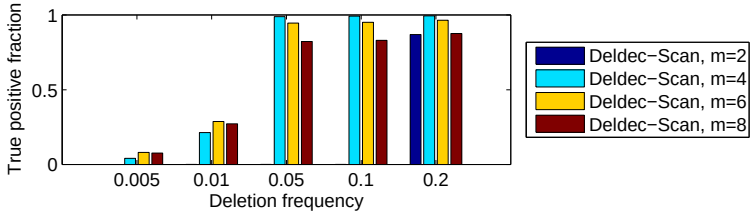
(a) 90 unrelated individuals



(b) 300 unrelated individuals

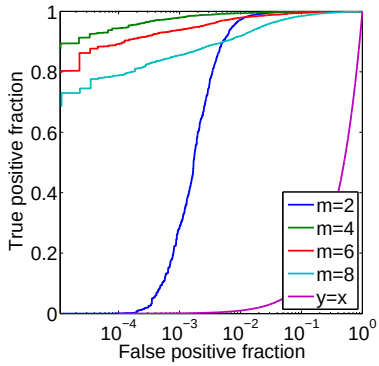


(c) 1500 unrelated individuals

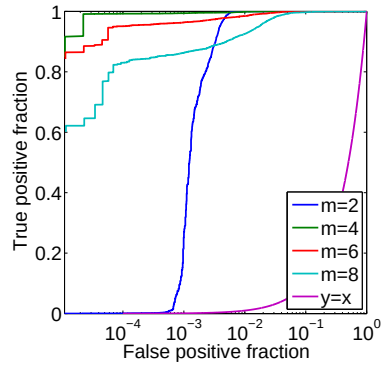


(d) 3000 unrelated individuals

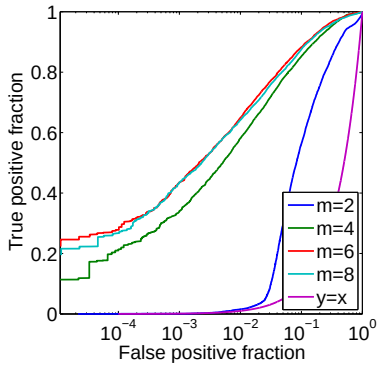
Figure 5.25: True positive fractions for Deldec-Scan under varying scenarios using unrelated data. The false positive fraction was fixed as 0.0001.



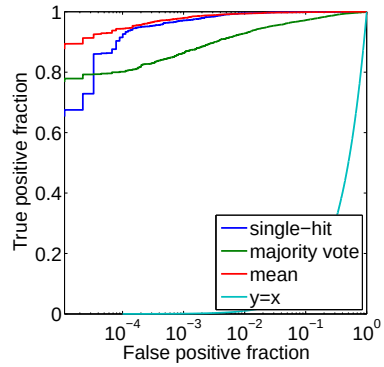
(a) The effect of window size m when the data set has 300 individuals, $f_0 = 0.1$ and the mean method is used.



(b) The effect of window size m when the data set has 3000 individuals, $f_0 = 0.1$ and the mean method is used.



(c) The effect of window size m when the data set has 3000 individuals, $f_0 = 0.01$ and the mean method is used.



(d) The effect of SNP deletion-status decision method when the data set has 300 individuals, $f_0 = 0.1$, window size m is 4 and the mean method is used.

Figure 5.26: Results for deletion detection with unrelated individuals as the data sets.

the SNPs are given in with respect to NCBI build 35. Note that this differs from the ones used for inversions, which used NCBI build 36 coordinates.

Both CEU and YRI data sets contained trios, so they were experimented using the Deldec EM algorithm for trios. For basic filtering of the data sets, SNPs that were genotyped at multiple genotyping centres were joined to their consensus genotype. If the called genotypes differed, that genotype call was set to no call. If a SNP had more than 20% no calls, the SNP was discarded.

Second, the number of apparent genotyping errors for each SNP was computed by counting the number Mendelian inconsistencies for each SNP. Only the cases in which both parents were homozygous and the child genotype did not have a null call were examined; if more than 20% of these examined cases in a SNP had a Mendelian inconsistency, the SNP was discarded.

Finally, all SNPs that were monoallelic or contained only heterozygotes together with no call genotypes were removed. In the end, the CEU data set has 3,280,460 SNPs and the YRI data set 3,463,706 SNPs. Both data sets have 30 trios.

To estimate the miscall parameter τ for the deletion model, we used the EM algorithm described in Section 4.4 that assumed SNPs to be independent. Applying this to all SNPs of even the shortest autosomal chromosome proved to be too time-consuming. Therefore, 3,000 SNPs were sampled at random from each chromosome and used to estimate the error rates for that chromosome.

Based on the results from the synthetic data sets, the moving window size for the deletion was set to 4 SNPs. The likelihood ratio test with the SNP-wise mean method was used to decide which SNPs were deleted; the p-value corresponding to the mean of the log-likelihood ratio had to be 10^{-10} or less for the SNP to be called deleted. Such an extreme significance threshold was used to decrease the number of false positives due to multiple testing.

To form an estimate of the deletion frequency, the mean estimated deletion frequency was computed for each SNP over the windows that contained the SNP. Once SNPs that were considered deleted were joined into contiguous candidate regions, the deletion frequency estimate for the region was the average of the SNP-wise averages.

The likelihood ratio test p-value is not reported, because the

value based on the likelihood ratio test was for all listed candidates very small, less than 10^{-16}. Reporting p-values smaller than this would not be of practical use.

To determine a p-value that might be more realistic than that of the likelihood ratio test value for the 4-SNP windows, the same filtering process as described above was performed to cleaned HapMap phase II data sets that had undergone a quality control process. The resulting data sets had 3,062,918 and 3,233,296 SNPs in CEU and YRI data sets, respectively. Because deletions appear as Mendelian inconsistencies and higher no call rates, these should have been effectively removed from the complete data set by the quality control process. Thus, the likelihood ratio test score histogram obtained from these data sets is closer to the null hypothesis. This is used to produce a p-value estimate $\hat{p}$. The estimate is conservative, because not all signal is removed from the data set. This p-value is reported as $\hat{p}$ in the result tables.

Because the methodology developed in Section 4.3.3 applies for unrelated data, the experiments included the joined JPT and CHB data sets. The data has 90 unrelated individuals. The filtering we did for the data was otherwise similar to that in the case of the YRI and CEU data sets, but the phase of computing the number of Mendelian inconsistencies was discarded, as these could not be estimated from this type of data. Therefore the results pertaining to this combined data set are less reliable than those of trio data sets, not necessarily because of the lack of data, but because of the unreliability of the data. The first, less filtered data set had 3,348,904 SNPs and the control data set (where a quality control process had been used before the filtering as described above) had 3,134,180 SNPs.

In Tables 5.15, 5.16 and 5.17 are reported only the 35 highest-scoring regions, as even with the selected p-value threshold there were far too many candidate regions to list. For the CEU data set, the experiments revealed 3,586 potential deletions. For the YRI and JPT+CHB data sets the numbers were 4,575 and 3,813, respectively.

The Database of Genetic Variants (July 2009 release, hg17 (NCBI build 35)) has 18,845 entries listed as CNVs and 3,540 indels. The version set in hg18 (NCBI build 36) coordinates had several more entries, but in reporting the overlaps only the build 35 version of

the database was used. The list of verified CNVs and indels were reported again even if only one basepair between a verified polymorphism and the candidate region was shared; in particular, an indel may have been only a few basepairs long and the candidate region several kilobases. Duplications may also have been erroneously reported below, as the types of the variations were not investigated more closely.

In many cases the candidate region intersected with CNVs or indels listed in several different articles. In the results, only the four most recent references were included, as in some cases there were several more references that reported intersecting regions. For all three data sets, a considerable proportion of the top-ranking regions is known to intersect with validated CNVs or indels. This strongly suggests that the haplotype frequency estimation method works well also in practice.

In many cases, the candidate region spanned a shorter region than the known CNVs. For instance, in the case of a proposed 33-kb deletion at chr2:52,663,182–52,696,603 in the CEU data set, the region overlapped variants reported in five articles. The longest variant was reported by Redon et al. [100] at 87 kb. The candidate region was completely contained within this reported variant. Also the regions reported by Kidd et al. [64], Cooper et al. [21] and de Smith et al. [26] contained all of the candidate region. However, the variants detected by McCarroll et al. [81, 82] are completely contained by the candidate region. The lengths of these variants are 31 kb and 30 kb.

On the other hand, we have candidates such as chr10:55,630,434–55,633,337 in the YRI data set. The candidate region is 3 kb long and the CNV detected by Pinto et al. [95] is 2.5 Mb in length. It is possible that this candidate is, in fact, a previously unknown short deletion instead of a part of a long CNV.

Overall, to summarize the reported 35 top-ranking regions in the CEU data set, there were 98 overlaps with reported variants. In 75 cases, the region listed in the Table 5.15 was contained inside the database entries. In 15 cases, the database entry was contained inside the candidate region, and in 8 cases, neither completely contained the other. The high number of candidate regions contained by the database entries can be due to several reasons. Because Deldec-Scan detects the signal from SNPs that are inside the dele-

Table 5.15: 35 highest-ranking candidate deletions from HapMap data, CEU data set. The coordinates are in NCBI build 35 coordinates.

Chr.	#SNPs	Start (bp)	Length (bp)	$\frac{\hat{p}}{10^{-7}}$	$\hat{f}_0$	Ref.
2	9	82,008,328	8,822	0	0.21	
3	11	111,145,471	4,973	0	0.30	
15	11	25,531,882	6,008	0	0.28	
2	13	52,663,182	33,421	19.6	0.30	[21, 26, 64, 82]
2	14	146,698,495	11,339	22.9	0.30	[20, 64, 81, 82]
2	16	21,313,283	3,349	26.1	0.29	
13	9	102,103,779	2,509	29.4	0.22	
2	10	34,618,159	29,985	29.4	0.28	[21, 26, 64, 82]
2	11	108,764,313	13,921	35.9	0.28	
3	20	164,039,007	39,931	35.9	0.27	[26, 64, 82, 142]
11	8	78,172,686	2,521	35.9	0.21	[59, 142]
6	11	67,098,871	1,915	39.2	0.19	[26, 61, 82, 106]
4	10	69,296,517	14,120	49.0	0.23	[20, 26, 82, 100]
10	10	53,680,643	3,397	55.5	0.25	
12	11	13,942,504	6,241	55.5	0.24	
12	7	69,810,558	1,773	55.5	0.17	
9	20	23,352,311	16,177	55.5	0.28	[26, 64, 82]
2	17	78,173,639	9,610	55.5	0.26	[95, 100, 137]
20	12	40,096,312	6,966	55.5	0.22	[95]
10	36	82,728,769	4,795	55.5	0.12	
3	9	84,591,358	1,597	65.3	0.14	
1	9	112,402,686	13,900	65.3	0.21	[20, 64, 82, 142]
5	6	150,895,003	2,924	65.3	0.26	
11	31	126,200,561	3,977	65.3	0.17	
10	7	49,799,777	1,559	65.3	0.17	[106]
10	9	14,501,432	1,873	65.3	0.23	
3	10	82,621,628	2,366	65.3	0.15	
8	12	39,449,314	2,095	65.3	0.19	[21, 26, 64, 82]
4	9	92,284,590	8,752	65.3	0.20	[20, 26, 64, 81]
2	16	106,336,720	7,781	65.3	0.26	[64, 82, 132]
6	10	6,532,297	1,215	65.3	0.25	
20	7	35,710,664	1,272	65.3	0.24	
10	9	53,490,273	4,449	65.3	0.23	[95]
2	8	235,786,624	3,411	68.6	0.19	
10	6	53,691,853	2,024	68.6	0.21	

Table 5.16: 35 highest-ranking candidate deletions from HapMap data, YRI data set. The coordinates are in NCBI build 35 coordinates.

Chr.	#SNPs	Start bp	Length (bp)	$\frac{\hat{p}}{10^{-7}}$	$\hat{f}_0$	Ref.
10	12	14,770,886	4,092	0	0.27	
6	21	6,527,373	6,139	0	0.23	
2	16	21,313,352	6,513	0	0.25	
11	8	28,850,004	1,127	0	0.20	
16	10	13,193,512	4,467	0	0.24	
10	9	55,630,434	2,903	0	0.26	[95]
4	14	70,378,886	12,279	0	0.26	[26, 77, 82, 100]
18	15	63,668,401	4,260	0	0.18	[106]
12	16	44,189,594	8,798	0	0.19	[21, 64, 82]
9	7	7,941,082	796	0	0.22	
9	13	102,187,316	5,743	0	0.31	
3	8	61,569,913	7,290	0	0.22	[100, 95]
13	6	38,832,462	3,399	0	0.17	[85]
8	7	3,286,222	1,963	0	0.23	
15	14	25,523,012	3,318	0	0.25	
11	9	5,828,695	3,748	0	0.18	[21, 61, 82, 106]
15	14	25,526,981	4,322	0	0.19	
20	6	8,824,090	1,821	0	0.21	
11	11	7,772,664	3,642	0	0.18	[21, 64, 82]
6	8	91,157,389	2,109	0	0.23	
2	8	43,531,971	7,340	0	0.24	
3	8	111,140,447	3,726	0	0.13	
7	6	88,512,780	2,565	3.1	0.16	[95, 106, 137]
4	6	115,535,846	7,287	3.1	0.22	[21, 64, 82, 106]
3	6	178,019,905	2,820	3.1	0.20	[95]
9	12	7,935,331	3,341	3.1	0.21	
2	6	226,792,000	3,294	3.1	0.17	
3	14	111,145,471	4,839	3.1	0.18	
6	8	133,065,721	5,939	3.1	0.22	
9	8	105,749,820	1,036	3.1	0.21	
15	14	32,521,237	8,788	3.1	0.19	[21, 61, 82, 106]
2	11	16,141,116	9,380	3.1	0.20	
6	11	106,930,170	2,731	3.1	0.23	
2	12	106,338,929	5,261	3.1	0.23	[64, 82, 132]
6	6	145,065,447	3,385	3.1	0.19	

Table 5.17: 35 highest-ranking candidate deletions from HapMap data, JPT+CHB data sets. The coordinates are in NCBI build 35 coordinates.

Chr.	#SNPs	Start bp	Length (bp)	$\frac{\hat{p}}{10^{-7}}$	$\hat{f}_0$	Ref.
3	10	3,452,526	9,907	0	0.27	[95]
5	9	150,162,906	3,401	0	0.26	[64, 100]
6	11	165,703,122	4,300	3.2	0.28	[64]
2	15	106,336,720	7,781	3.2	0.27	[64, 82, 132]
6	22	32,759,846	1,517	3.2	0.25	[26, 61, 106, 142]
11	11	126,196,008	2,012	3.2	0.27	
12	10	60,080,170	3,154	3.2	0.26	[100]
2	15	67,143,097	6,649	3.2	0.17	
22	8	37,688,911	7,021	3.2	0.25	[21, 26, 64, 82]
2	12	5,881,785	4,690	3.2	0.19	[142]
11	16	126,201,806	2,658	3.2	0.26	
2	10	64,430,563	6,414	3.2	0.19	[100, 95]
5	17	145,293,466	6,309	3.2	0.21	
2	9	33,971,663	3,474	3.2	0.19	[61]
5	10	122,505,940	1,895	6.4	0.25	
14	9	82,370,501	1,522	9.6	0.28	
2	11	39,933,925	6,181	9.6	0.15	
5	7	170,945,536	1,434	16.0	0.15	
2	19	5,756,171	7,539	16.0	0.23	[106, 142]
9	14	73,439,209	5,734	16.0	0.26	
1	7	112,404,201	3,515	16.0	0.21	[64, 100, 132, 142]
5	7	131,755,245	1,366	16.0	0.20	
3	10	191,221,744	2,411	16.0	0.28	[26, 81, 106]
5	7	170,942,438	1,666	16.0	0.17	
3	12	74,817,834	7,623	16.0	0.19	
21	7	40,804,037	1,926	16.0	0.26	[85]
22	6	37,682,537	1,751	16.0	0.21	[26, 64, 132]
5	6	145,282,859	3,201	16.0	0.25	
20	7	50,767,082	5,161	16.0	0.20	[59]
2	16	78,174,297	8,685	16.0	0.27	[95, 100, 137]
13	6	64,221,390	1,902	16.0	0.25	[105]
8	9	87,149,323	3,396	19.1	0.21	
10	12	16,869,213	2,718	19.1	0.16	
9	6	105,749,820	2,473	19.1	0.20	
11	15	4,925,512	5,845	22.3	0.22	[26, 64, 82, 100]

tion, the estimated deletion should, ideally, be at most as long as the actual deletion. Another possible cause is that the validation techniques used to discover the entries that were entered into the database may have overestimated the length of the deletion. Finally, the used LRT statistic threshold for considering a SNP deleted may have been set too high, which would have cut the ends of the underlying polymorphisms from the candidate set.

The majority of the candidates at the end of the list were one-SNP long. Of the 35 candidate regions listed by Deldec-Scan as potential deletions in the CEU data set with the worst p-values, 16 intersected with DGV CNV or indel entries. This also hints at the possibility that the selected threshold for likelihood ratio test was set too high for the majority of the deleted SNPs to be detected. Nonetheless, the threshold was left high to reduce the number of false positives.

If the same threshold was used solely on the results on QC-passed genotype data, CEU, YRI and JPT+CHB data would have given us 653, 523 and 1033 candidates, respectively. Of the top 35 candidates, 12, 10 and 6 in the same populations intersected with known CNVs or indels.

5.3.4 Discussion

Although Deldec-Scan is not as good as microdel for detecting deletions in trio data in most scenarios based on the results on simulated data, it is nonetheless a valid option for detecting them in unrelated data, which is something that microdel is not suited for. The estimation of the error parameters is not robust, as was evidenced by Figure 5.24. A more rigorous approach would likely improve the performance. It is unclear how to accomplish this.

It might be possible to improve the end-point detection of deletions. The presented methods are heuristics utilizing data only by the estimated likelihood ratio test scores. The microdel program estimated deleted haplotypes in the data set and used these to decide on where the deletion ends. A similar approach might work with Deldec-Scan as well.

Finally, the deletion candidate list of Kohler and Cutler [68] was nearly filled with previously known CNVs. By comparison, the list of previously validated CNVs in the candidate deletion lists in Sec-

tion 5.3.3 appears very sparse. One possibility is that the used data screening was not as efficient as that of Kohler and Cutler in removing false signal, although the same effect (that only some candidate deletions had literature references) is seen in Deldec-Scan results for HapMap's QC-filtered data sets as well. Another possibility is that the change in the used data set (HapMap phase II compared to HapMap phase I used in [68]) caused this, because the larger phase II data set could provide greater resolution in regions where there were deletion polymorphisms.

CHAPTER 6

Discussion

The aim of this thesis was to present novel – or mostly so – methods to detect genetic structural variation from SNP data. The presented algorithm for detecting inversions, NBS-Scan, utilized the signal resulting from the decreased gene flow between different chromosome arrangements. Deldec-Scan, an algorithm for detecting deletions, is based on a previous algorithm by Corona et al. [22]. There still remains much to do in the field of using SNP data to discover inversions or deletions. Technological advances in, e.g., resequencing are however in the near future a possible reason why such research could lose some of its relevance.

For now, let us assume that analyzing SNP data to detect structural variation remains a relevant topic also in the future. As mentioned in Chapter 3, there are at least two types of signal that may reveal the presence of inversions: the linkage disequilibrium (LD) patterns near the inversion breakpoints and the decreased gene flow between the two arrangements, which should lead to the four-field pattern discussed in Section 3.1. There now exist methods that attempt to detect either one of the signals. But can both signals be detected together to result in better detection accuracy? As seen in Section 5.2.4, the gene flow signal is not always present even with frequent inversions. There may also be other, possibly better ways of detecting inversions from SNP data, such as principal component analysis as suggested by Deng et al. [28]. The simple experiment of taking the intersection of the inversions predicted by Sindi and Raphael's [112] and NBS-Scan in the HapMap data set showed that combining the results of the two methods might sometimes be use-

ful.

The idea of using lowered recombination rates in inversion detection came originally from observing the 900-kb inversion in chromosome 17. This inversion seems to remain the best fit for the four-field pattern. The clear and visible signal sparked the idea of using the division to detect inversions even in less clear cases. It appeared that this approach does not generalize to all known inversions.

Antonacci et al. [4] deduced that some of the six inversions investigated in their article occurred on at least two different haplotype backgrounds. This is in clear disagreement with the assumption of inversion uniqueness done by NBS and InvCoal and is a possible reason for the poor performance of the two.

The algorithms presented in this thesis, NBS-Scan and Deldec-Scan, are similar yet different. Both are based on the idea of a fixed-width window moving over the data, after which these windows are joined together. The algorithms differ in how these windows are joined together and how the window move over the chromosomes. It is reasonable to ask why were the algorithms not made more similar in this respect. This is mostly due to the sizes of the windows these algorithms cover and how much signal a single SNP can carry in identifying structural variants. For finding deletions, already 4-SNP windows are sufficient for identifying variants. It is possible, at least with some level of approximation, to estimate in which proportions to divide the signal among the covered SNPs. This is not the case with normalized bicomponent score (NBS) introduced here for detecting inversions.

For detecting deletions, there already are multiple different methods, some of which are applicable even before calling the genotypes. As it is, the approach by estimating haplotype frequencies might not appear promising. However, maybe the largest problem that prevents Deldec-Scan from performing at a level comparable to microdel is the difficulty in determining the deletion end-point accurately. However, Deldec-Scan can work also in the case of unrelated individuals whereas microdel cannot. For a deletion spanning multiple SNPs, the problem is not in detecting the deletion in general but in determining where the deletion ends. There might be better approaches to this problem beside the simple methods discussed in Section 4.6.

On the topic of investigating the performance of inversion-detection algorithms, the development of a coalescent simulator that can simulate inversions to some level of accuracy was a small milestone on the road to the completion of this thesis. The InvCoal simulator serves as a starting point for future development. There are several ways to improve the simulation scheme presented in Chapter 2, such as varying recombination rate within the simulated segment and stricter adherence to reality in the inversion and recombination models.

Perhaps the most pressing concern, however, is the modelling of population history. The problem of selecting a realistic model was not addressed in this thesis. One question is whether the joint effective population size of the two subpopulations should be considered a constant or not. Another is selecting a reasonable model to depict the size changes in the haplotype population of the new arrangement.

The simulator output was not comprehensively compared to known inversions in the human genome outside an examination of how NBS behaves inside and outside inversions in Section 5.1. This is a shortcoming because few of the inversions in the HapMap data set are known to resemble the simulator output. There are several potential reasons for this discrepancy beside the population history model. First, the simulator does not model the current knowledge of the effect inversions have on the human genome. In particular, gametogenesis, or the process of generating gametes, differs between species, and what is true for *Drosophila* might not hold true for humans when inversions and recombinations are in question. These differences include the absence of spontaneous meiotic recombinations in *Drosophila* males (e.g., [51]). InvCoal does not attempt to model either organism completely. Second, the author's current knowledge in this respect is insufficient to accurately model inversions. Finally, the measuring and identification of SNPs within the inversion may have produced some errors in the data set or the inversions may have affected the haplotype inference process. One potential future avenue for research would be to identify the source of this discrepancy between simulator output and HapMap data.

Overall, the experimental results for algorithms NBS-Scan and Deldec-Scan were promising. Both methods detected several previously known polymorphisms while providing also an ample number

of novel candidate regions for experimental validation. On real human genome data, however, the method of Sindi and Raphael [112] seemed to outperform NBS-utilizing detection schemes in most cases.

The use of only tag-SNPs notably decreased the power of NBS-Scan. This is unfortunate for the algorithm's application in genome-wide association studies, as the studies genotype only a set of representative SNPs across the genome and thus only tag-SNPs would be available in such studies. Still, if the regions found associated with the interesting phenotype in a genome-wide association study are investigated by resequencing or genotyping more SNPs in these regions, structural variants can be discovered. The effect of tag-SNP algorithms on other algorithms for detecting inversions appears not to have been previously investigated; this is an interesting topic for future studies.

References

[1] S.-M. Ahn, T.-H. Kim, S. Lee, D. Kim, H. Ghang, et al. The first Korean genome sequence and analysis: Full genome sequencing for a socio-ethnic group. *Genome Research*, 19:1622–1629, 2009.

[2] S. M. Aji and McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46:325–343, 2000.

[3] S. E. Alter, E. Rynes, and S. R. Palumbi. DNA evidence for historic population size and past ecosystem impacts of gray whales. *PNAS*, 104:15162–15167, 2007.

[4] F. Antonacci, J. M. Kidd, T. Marques-Bonet, M. Ventura, P. Siswara, et al. Characterization of six human disease-associated inversion polymorphisms. *Human Molecular Genetics*, 18:2555–2566, 2009.

[5] N. Arnheim, P. Calabrese, and I. Tiemann-Boege. Mammalian meiotic recombination hot spots. *Annual Review of Genetics*, 41:369–399, 2007.

[6] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[7] J. E. Atkins, E. G. Boman, and B. Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal of Computing*, 28:297–310, 1998.

[8] N. D. Avent, P. G. Martin, S. S. Armstrong-Fisher, W. Liu, K. M. Finning, et al. Evidence of genetic diversity underlying Rh D-, weak D (Du), and partial D phenotypes as determined by multiplex polymerase chain reaction analysis of the *RHD* gene. *Blood*, 89:2568–2577, 1997.

[9] V. Bansal, A. Bashir, and V. Bafna. Evidence for large inversion polymorphisms in the human genome from HapMap data. *Genome Research*, 17:219–230, 2007.

[10] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of Royal Statistical Society: Series B*, 57:289–300, 1995.

[11] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456:53–59, 2008.

[12] L. E. Berchowitz and G. P. Copenhaver. Genetic interference: Don't stand so close to me. *Current Genomics*, 11:91–102, 2010.

[13] N. Bosch, M. Morell, I. Ponsa, J. M. Mercader, L. Armengol, and X. Estivill. Nucleotide, cytogenetic and expression impact of the human chromosome 8p23.1 inversion polymorphism. *PLoS ONE*, 4:e8269, 2009. doi:10.1371/journal.pone.0008269.

[14] J. M. Braverman, R. R. Hudson, N. L. Kaplan, C. H. Langley, and W. Stephan. The hitchhiking effect on the site frequency spectrum of DNA polymorphisms. *Genetics*, 140:783–796, 1995.

[15] K. W. Broman and J. L. Weber. Characterization of human crossover interference. *American Journal of Human Genetics*, 66:1911–1926, 2000.

[16] T. C. Bruen, P. Hervé, and D. Bryant. A simple and robust statistical test for detecting the presence of recombination. *Genetics*, 172:2665–2681, 2006.

[17] C. S. Carlson, M. A. Eberle, M. J. Rieder, Q. Yi, L. Kruglyak, and D. A. Nickerson. Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. *American Journal of Human Genetics*, 74:106–120, 2004.

[18] A. Chovnik. Gene conversion and transfer of genetic information within the inverted region of inversion heterozygotes. *Genetics*, 75:123–131, 1973.

[19] A. G. Clark, M. J. Hubisz, C. D. Bustamante, S. H. Williamson, and R. Nielsen. Ascertainment bias in studies of human genome-wide polymorphism. *Genome Research*, 15:1496–1502, 2005.

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

[21] G. M. Cooper, T. Zerr, J. M. Kidd, E. E. Eichler, and D. A. Nickerson. Systematic assessment of copy number variant detection via genome-wide SNP genotyping. *Nature Genetics*, 40:1193–1203, 2008.

[22] E. Corona, B. Raphael, and E. Eskin. Identification of deletion polymorphisms from haplotypes. In T. Speed and H. Huang, editors, *Research in Computational Molecular Biology*, volume 4453 of *LNBI*, pages 354–365. Springer, 2007.

[23] J. A. Coyne, W. Meyers, A. P. Crittenden, and P. Sniegowski. The fertility effects of pericentric inverions in *Drosophila melanogaster*. *Genetics*, 134:487–496, 1993.

[24] E. M. Cross and W. W. Chaffin. Use of the binomial theorem in interpreting results of multiple tests of significance. *Educational and Psychological Measurement*, 42:25–34, 1982.

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

[26] A. J. de Smith, A. Tsalenko, N. Sampas, A. Scheffer, N. A. Yamada, et al. Array CGH analysis of copy number variation identifies 1284 new genes variant in healthy white males: implications for association studies of complex diseases. *Human Molecular Genetics*, 16:2783–2794, 2007.

[27] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39:1–38, 1977.

[28] L. Deng, Y. Zhang, J. Kang, T. Liu, H. Zhao, et al. An unusual haplotype structure on human chromosome 8p23 derived from the inversion polymorphism. *Human Mutation*, 29:1209–1216, 2008.

[29] B. Devlin and N. Risch. A comparison of linkage disequilibrium measures for fine-scale mapping. *Genomics*, 29:311–322, 1995.

[30] M. P. Donnelly, P. Peristera, E. Grigorenko, D. Gurwitz, S. Q. Mehdi, et al. The distribution and most recent common ancestor of the 17q21 inversion in humans. *American Journal of Human Genetics*, 86:161–171, 2010.

[31] P. Donnelly and S. Tavaré. Coalescents and genealogical structure under neutrality. *Annual Review of Genetics*, 29:401–421, 1995.

[32] W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics: An Introduction*. Springer Science+Business Media, New York, second edition, 2005.

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

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

[35] P. Fearnhead and P. Donnelly. Estimating recombination rates from population genetic data. *Genetics*, 159:1299–1318, 2001.

[36] L. Feuk, A. R. Carson, and S. W. Scherer. Structural variation in the human genome. *Nature Reviews Genetics*, 7:85–97, 2006.

[37] R. A. Fisher. *Genetical theory of natural selection*. Dover, New York, 2. rev. edition, 1958.

[38] M. Fortelius, A. Gionis, J. Jernvall, and H. Mannila. Spectral ordering and biochronology of European fossil mammals. *Paleobiology*, 32:206–214, 2006.

[39] E. Foss, R. Lande, F. W. Stahl, and C. M. Steinberg. Chiasma interference as a function of genetic distance. *Genetics*, 133:681–691, 1993.

[40] L. Franke, C. G. de Kovel, Y. S. Aulchenko, G. Trynka, A. Zhernakova, et al. Detection, imputation, and association analysis of small deletions and null alleles on oligonucleotide arrays. *American Journal of Human Genetics*, 82:1316–1333, 2008.

[41] S. B. Gabriel, S. F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, et al. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.

[42] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, second edition, 2004.

[43] R. C. Griffiths and P. Marjoram. Ancestral inference from samples of DNA sequences with recombination. *Journal of Computational Biology*, 3:479–502, 1996.

[44] J. B. S. Haldane. The combination of linkage values, and the calculation of distances between the loci of linked factors. *Journal of Genetics*, 8:299–309, 1919.

[45] E. Halperin and E. Eskin. Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*, 20:1842–1849, 2004.

[46] E. Halperin, G. Kimmel, and R. Shamir. Tag SNP selection in genotype data for maximizing SNP prediction accuracy. *Bioinformatics*, 21:i195–i203, 2005.

[47] G. H. Hardy. Mendelian proportions in a mixed population. *Science*, 28:49–50, 1908.

[48] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction.* Springer-Verlag, New York, 2001.

[49] J. Hein, M. H. Schierup, and C. Wiuf. *Gene Genealogies, Variation and Evolution: a primer in coalescent theory.* Oxford University Press, 2005.

[50] D. A. Hinds, L. L. Stuve, G. B. Nilsen, E. Halperin, E. Eskin, et al. Whole-genome patterns of common DNA variation in three human populations. *Science*, 307:1072–1079, 2005.

[51] Y. Hiraizumi, B. Slatko, C. Langley, and A. Nill. Recombination in drosophila melanogaster male. *Genetics*, 73:439–444, 1973.

[52] A. A. Hoffmann and L. H. Rieseberg. Revisiting the impact of inversions in evolution: From population genetic markers to drivers of adaptive shifts and speciation? *Annual Review of Ecology, Evolution and Systematics*, 39:21–42, 2008.

[53] A. A. Hoffmann, C. M. Sgrò, and A. R. Weeks. Chromosomal inversion polymorphisms and adaptation. *Trends in Ecology and Evolution*, 19:482–488, 2004.

[54] E. A. Housworth and F. W. Stahl. Crossover interference in humans. *American Journal of Human Genetics*, 73:188–197, 2003.

[55] R. R. Hudson. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23:183–201, 1983.

[56] R. R. Hudson. Two-locus sampling distributions and their applications. *Genetics*, 159:1805–1817, 2001.

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

[58] R. R. Hudson and N. L. Kaplan. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, 111:147–164, 1985.

[59] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, et al. Detection of large-scale variation in the human genome. *Nature Genetics*, 36:949–51, 2004.

[60] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[61] A. Itsara, G. M. Cooper, C. Baker, S. Girirajan, J. Li, et al. Population analysis of large copy number variants and hotspots of human genetic disease. *American Journal of Human Genetics*, 84:148–161, 2009.

[62] A. J. Jeffreys, L. Kauppi, and R. Neumann. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nature Genetics*, 29:217–222, 2001.

[63] L. Kauppi, A. J. Jeffreys, and S. Keeney. Where the crossovers are: recombination distribution in mammals. *Nature Reviews Genetics*, 5:413–424, 2004.

[64] J. M. Kidd, G. M. Cooper, W. F. Donahue, H. S. Hayden, N. Sampas, et al. Mapping and sequencing of structural variation from eight human genomes. *Nature*, 453:56–64, 2008.

[65] J. F. C. Kingman. The coalescent. *Stochastic Processes and their Applications*, 13:235–248, 1982.

[66] M. Kirkpatrick and N. Barton. Chromosome inversions, local adaptation and speciation. *Genetics*, 173:419–434, 2006.

[67] D. E. Knuth. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*. Addison-Wesley, Reading, Massachusetts, Third edition, 1997.

[68] J. R. Kohler and D. J. Cutler. Simultaneous discovery and testing of deletions for disease association in SNP genotyping studies. *American Journal of Human Genetics*, 81:684–699, 2007.

[69] M. Koivisto. *Sum-Product Algorithms for the Analysis of Genetic Risks*. PhD thesis, University of Helsinki, January 2004.

[70] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, et al. Paired-end mapping reveals extensive structural variation in the human genome. *Science*, 318:420–426, 2007.

[71] F. R. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[72] M. K. Kuhner. Coalescent genealogy samplers: windows into population history. *Trends in Ecology and Evolution*, 24:86–93, 2008.

[73] M. K. Kuhner, J. Yamato, and J. Felsenstein. Maximum likelihood estimation of recombination rates from population data. *Genetics*, 156:1393–1401, 2000.

[74] S. Levy, G. Sutton, P. C. Ng, L. Feuk, A. L. Halpern, et al. The diploid genome sequence of an individual human. *PLoS Biology*, 5:e254, 2007. doi:10.1371/journal.pbio.0050254.

[75] S. Lin, A. Chakravarti, and D. J. Cutler. Exhaustive allelic transmission disequilibrium tests as a new approach to genome-wide association studies. *Nature Genetics*, 36:1181–1188, 2004.

[76] S. Lin and T. P. Speed. Incorporating crossover interference into pedigree analysis using the χ^2 model. *Human Heredity*, 46:315–322, 1996.

[77] D. P. Locke, A. J. Sharp, S. A. McCarroll, S. D. McGrath, T. L. Newman, et al. Linkage disequilibrium and heritability of copy-number polymorphisms within duplicated regions of the human genome. *American Journal of Human Genetics*, 79:275–290, 2006.

[78] J. R. Lupski and P. Stankiewicz. Genomic disorders: molecular mechanisms for rearrangements and conveyed phenotypes. *PLoS Genetics*, 1:e49, 2005. doi:10.1371/journal.pgen.0010049.

[79] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

[80] J. Marchini, D. Cutler, N. Patterson, M. Stephens, E. Eskin, et al. A comparison of phasing algorithms for trios and unrelated individuals. *American Journal of Human Genetics*, 78:437–450, 2006.

[81] S. A. McCarroll, T. N. Hadnott, G. H. Perry, P. C. Sabeti, M. C. Zody, et al. Common deletion polymorphisms in the human genome. *Nature Genetics*, 38:86–92, 2006.

[82] S. A. McCarroll, F. G. Kuruvilla, J. M. Korn, S. Cawley, J. Nemesh, et al. Integrated detection and population-genetic analysis of SNPs and copy number variation. *Nature Genetics*, 40:1166–1174, 2008.

[83] M. S. McPeek and T. P. Speed. Modeling interference in genetic recombination. *Genetics*, 139:1031–1044, 1995.

[84] G. McVean, P. Awadalla, and P. Fearnhead. A coalescent-based method for detecting and estimating recombination from gene sequences. *Genetics*, 160:1231–1241, 2002.

[85] R. E. Mills, C. T. Lutting, C. E. Larkins, A. Beauchamp, C. Tsui, et al. An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Research*, 16:1182–1190, 2006.

[86] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge Univ Press, Cambridge, 2005.

[87] A. Munté, J. Rozas, M. Aguadé, and C. Segarra. Chromosomal inversion polymorphism leads to extensive genetic struc-

ture: A multilocus survey in *Drosophila subobscura*. *Genetics*, 169:1573–1581, 2005.

[88] A. Navarro, A. Barbadilla, and A. Ruiz. Effect of inversion polymorphism on the neutral nucleotide variability of linked chromosomal regions in Drosophila. *Genetics*, 155:685–698, 2000.

[89] A. Navarro, E. Betrán, A. Barbadilla, and A. Ruiz. Recombination and gene flux caused by gene conversion and crossing over in inversion heterokaryotypes. *Genetics*, 146:695–709, 1997.

[90] R. Nielsen. Estimation of population parameters and recombination rates from single nucleotide polymorphisms. *Genetics*, 154:931–942, 2000.

[91] M. Nordborg. Coalescent theory. In D. Balding, M. Bishop, and C. Cannings, editors, *Handbook of Statistical Genetics*, pages 179–212. John Wiley and Sons Ltd., Chichester, UK, 2001.

[92] P. F. O'Reilly, L. J. M. Coin, and C. J. Hoggart. invertFREGENE: software for simulating inversions in population genetic data. *Bioinformatics*, 26:838–840, 2010.

[93] N. Patil, A. J. Berno, D. A. Hinds, W. Barrett, J. M. Doshi, et al. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

[94] B. A. Pierce. *Genetics: A Conceptual Approach*. W.H. Freeman and Company, New York, Second Edition, 2005.

[95] D. Pinto, C. Marshall, L. Feuk, and S. W. Scherer. Copy-number variation in control population cohorts. *Human Molecular Genetics*, 16:R168–R173, 2007.

[96] K. D. Pruitt, T. Tatusova, and D. R. Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleid Acids Research*, 35(Database issue):D61–D65, 2007.

[97] Psychiatric GWAS Consortium Coordinating Committee. Genomewide association studies: History, rationale, and prospects for psychiatric disorders. *American Journal of Psychiatry*, 166:540–556, 2009.

[98] J. M. Ranz, F. Casals, and A. Ruiz. How malleable is the eukaryotic genome? Extreme rate of chromosomal rearrangement in the genus Drosophila. *Genome Research*, 11:230–239, 2001.

[99] J. M. Ranz, D. Maurin, Y. S. Chan, M. von Grotthuss, L. W. Hillier, et al. Principles of genome evolution in the *Drosophila melanogaster* species group. *PLoS Biology*, 5:e152, 2007. doi:10.1371/journal.pbio.0050152.

[100] R. Redon, S. Ishikawa, K. R. Fitch, L. Feuk, G. H. Perry, et al. Global variation in copy number in the human genome. *Nature*, 444:444–454, 2006.

[101] D. E. Reich, M. Cargill, S. Bolk, J. Ireland, P. C. Sabeti, et al. Linkage disequilibrium in the human genome. *Nature*, 411:199–204, 2001.

[102] L. Rieseberg. Chromosomal rearrangements and speciation. *Trends in Ecology & Evolution*, 16:351–358, 2001.

[103] S. W. Schaeffer and W. W. Anderson. Mechanisms of genetic exchange within the chromosomal inversions of *Drosophila pseudoobscura*. *Genetics*, 171:1729–1739, 2005.

[104] S. F. Schaffner, C. Foo, S. Gabriel, D. Reich, M. J. Daly, and D. Altshuler. Calibrating a coalescent simulation of human genome sequence variation. *Genome Research*, 15:1576–1583, 2005.

[105] J. Sebat, B. Lakshmi, J. Troge, J. Alexander, J. Young, et al. Large-scale copy number polymorphism in the human genome. *Science*, 305:525–528, 2004.

[106] T. H. Shaikh, X. Gai, J. C. Perin, J. T. Glessner, H. Xie, et al. High-resolution mapping and analysis of copy number variations in the human genome: a data resource for clinical

and research applications. *Genome Research*, 19:1682–1690, 2009.

[107] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[108] B. Shapiro, A. J. Drummond, A. Rambaut, M. C. Wilson, P. E. Matheus, et al. Rise and fall of the Beringian steppe bison. *Science*, 306:1561–1565, 2004.

[109] A. J. Sharp, Z. Cheng, and E. E. Eichler. Structural variation of the human genome. *Annual Review of Genomics and Human Genetics*, 7:407–442, 2006.

[110] C. J. Shaw and J. R. Lupski. Implications of human genome architecture for rearrangement-based disorders: the genomic basis of disease. *Human Molecular Genetics*, 13:R57–R64, 2004.

[111] S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29:308–311, 2001.

[112] S. S. Sindi and B. J. Raphael. Identification and frequency estimation of inversion polymorphisms from haplotype data. In S. Batzoglou, editor, *Research in Computational Molecular Biology*, volume 5541 of *LNBI*, pages 418–433. Springer, 2009.

[113] M. Slatkin. Linkage disequilibrium — understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9:477–485, 2008.

[114] C. C. A. Spencer, Z. Su, P. Donnelly, and J. Marchini. Designing genome-wide association studies: Sample size, power, imputation, and the choice of genotyping chip. *PLoS Genetics*, 5:e1000477, 2009. doi:10.1371/journal.pgen.100477.

[115] D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421:284–305, 2007.

[116] R. E. Stearns and H. B. Hunt III. An algebraic model for combinatorial problems. *SIAM Journal on Computing*, 25(2):448–476, 1996.

[117] H. Stefansson, A. Helgason, G. Thorleifsson, V. Steinthorsdottir, G. Masson, et al. A common inversion under selection in Europeans. *Nature Genetics*, 37:129–137, 2005.

[118] J. Storey. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics*, 31:2013–2035, 2003.

[119] J. D. Storey. A direct approach to false discovery rates. *Journal of Royal Statistical Society: Series B*, 64:479–498, 2002.

[120] C. Strobeck. Average number of nucleotide differences in a sample from a single subpopulation: a test for population subdivision. *Genetics*, 117:149–153, 1987.

[121] M. P. Stumpf and G. A. McVean. Estimating recombination rates from population-genetic data. *Nature Reviews Genetics*, 4:959–968, 2003.

[122] A. H. Sturtevant. Genetic factors affecting the strength of linkage in *Drosophila*. *Proceedings of the National Academy of Science*, 3:555–558, 1917.

[123] J. A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

[124] F. Tajima. Statistical method for testing the neutral mutation hypothesis by dna polymorphism. *Genetics*, 123:585–595, 1989.

[125] N. Takahata. Allelic genealogy and human evolution. *Molecular Biology and Evolution*, 10:2–22, 1993.

[126] A. Tenesa, P. Navarro, B. J. Hayes, D. L. Duffy, G. M. Clarke, et al. Recent human effective population size estimated from linkage disequilibrium. *Genome Research*, 17:520–526, 2007.

[127] The International HapMap Consortium. The International HapMap Project. *Nature*, 426:467–475, 2003.

[128] The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437:1299–1320, 2005.

[129] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449:851–861, 2007.

[130] N. S. Thomas, V. Bryant, V. Maloney, A. E. Cockwell, and P. A. Jacobs. Investigation of the origins of human autosomal inversions. *Human Genetics*, 123:607–616, 2008.

[131] P. Turnpenny and S. Ellard. *Emery's Elements of Medical Genetics*. Elsevier, twelfth edition, 2005.

[132] E. Tuzun, A. J. Sharp, J. A. Bailey, K. Rajinder, V. A. Morrison, et al. Fine-scale structural variation of the human genome. *Nature Genetics*, 37:727–732, 2005.

[133] C. Varin. On composite marginal likelihoods. *Advances in Statistical Analysis*, 92:1–28, 2008.

[134] J. C. Venter, M. Adams, E. Myers, P. Li, R. Mural, et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.

[135] L. V. Wain, J. A. L. Armour, and M. D. Tobin. Genomic copy number variation, human health, and disease. *Lancet*, 374:340–350, 2009.

[136] J. Wang, W. Wang, R. Li, Y. Li, G. Tian, et al. The diploid genome sequence of an Asian individual. *Nature*, 456:60–65, 2008.

[137] K. Wang, M. Li, D. Hadley, R. Liu, J. Glessner, et al. PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research*, 17:1665–1674, 2007.

[138] W. Weinberg. Über den Nachweis der Vererbung beim Menschen. *Jahreshefte des Vereins für vaterländische Naturkunde in Württemberg*, 64:368–382, 1908.

[139] D. A. Wheeler, M. Srinivasan, M. Egholm, Y. Shen, L. Chen, et al. The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452:872–876, 2008.

[140] S. S. Wilks. *Mathematical Statistics*. John Wiley & Sons Inc, New York, 1962.

[141] C. Wiuf and J. Hein. The coalescent with gene conversion. *Genetics*, 155:451–462, 2000.

[142] K. K. Wong, R. J. deLeeuw, N. S. Dosanjh, L. Kimm, Z. Cheng, et al. A comprehensive analysis of common copy-number variations in the human genome. *American Journal of Human Genetics*, 80:91–104, 2007.

[143] S. Wright. Evolution in Mendelian populations. *Genetics*, 16:97–159, 1931.

[144] F.-I. Yamamoto, H. Clausen, T. White, J. Marken, and S.-I. Hakomori. Molecular genetic basis of the histo-blood group ABO system. *Nature*, 345:229–233, 1990.

[145] F. Yates. The design and analysis of factorial experiments. *Harpenden: Imperial Bureau of Soil Science Technical Communication 35*, 1937.

[146] J. Zhang, W. L. Rowe, A. G. Clark, and K. H. Buetow. Genomewide distribution of high-frequency, completely mismatching SNP haplotype pairs observed to be common across human populations. *American Journal of Human Genetics*, 73:1073–1081, 2003.

[147] G. Zou and H. Zhao. Haplotype frequency estimation in the presence of genotyping errors. *Human Heredity*, 56:131–138, 2003.

[148] S. Zöllner and A. von Haeseler. A coalescent approach to study linkage disequilibrium between single-nucleotide polymorphisms. *American Journal of Human Genetics*, 66:615–628, 2000.

TIETOJENKÄSITTELYTIETEEN LAITOS
PL 68 (Gustaf Hällströmin katu 2 b)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Gustaf Hällströmin katu 2 b)
FIN-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports may be ordered from: Kumpula Science Library, P.O. Box 64, FIN-00014 University of Helsinki, FINLAND.

A-2007-5 S. Leggio: A Decentralized Session Management Framework for Heterogeneous Ad-Hoc and Fixed Networks. 230 pp. (Ph.D. Thesis)

A-2007-6 O. Riva: Middleware for Mobile Sensing Applications in Urban Environments. 195 pp. (Ph.D. Thesis)

A-2007-7 K. Palin: Computational Methods for Locating and Analyzing Conserved Gene Regulatory DNA Elements. 130 pp. (Ph.D. Thesis)

A-2008-1 I. Autio: Modeling Efficient Classification as a Process of Confidence Assessment and Delegation. 212 pp. (Ph.D. Thesis)

A-2008-2 J. Kangasharju: XML Messaging for Mobile Devices. 24+255 pp. (Ph.D. Thesis).

A-2008-3 N. Haiminen: Mining Sequential Data – in Search of Segmental Structures. 60+78 pp. (Ph.D. Thesis)

A-2008-4 J. Korhonen: IP Mobility in Wireless Operator Networks. (Ph.D. Thesis)

A-2008-5 J.T. Lindgren: Learning nonlinear visual processing from natural images. 100+64 pp. (Ph.D. Thesis)

A-2009-1 K. Hätönen: Data mining for telecommunications network log analysis. 153 pp. (Ph.D. Thesis)

A-2009-2 T. Silander: The Most Probable Bayesian Network and Beyond. (Ph.D. Thesis)

A-2009-3 K. Laasonen: Mining Cell Transition Data. 148 pp. (Ph.D. Thesis)

A-2009-4 P. Miettinen: Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms. 164+6 pp. (Ph.D. Thesis)

A-2009-5 J. Suomela: Optimisation Problems in Wireless Sensor Networks: Local Algorithms and Local Graphs. 106+96 pp. (Ph.D. Thesis)

A-2009-6 U. Köster: A Probabilistic Approach to the Primary Visual Cortex. 168 pp. (Ph.D. Thesis)

A-2009-7 P. Nurmi: Identifying Meaningful Places. 83 pp. (Ph.D. Thesis)

A-2009-8 J. Makkonen: Semantic Classes in Topic Detection and Tracking. 155 pp. (Ph.D. Thesis)

A-2009-9 P. Rastas: Computational Techniques for Haplotype Inference and for Local Alignment Significance. 64+50 pp. (Ph.D. Thesis)

A-2009-10 T. Mononen: Computing the Stochastic Complexity of Simple Probabilistic Graphical Models. 60+46 pp. (Ph.D. Thesis)

A-2009-11 P. Kontkanen: Computationally Effcient Methods for MDL-Optimal Density Estimation and Data Clustering. 75+64 pp. (Ph.D. Thesis)

A-2010-1 M. Lukk: Construction of a global map of human gene expression - the process, tools and analysis. 120 pp. (Ph.D. Thesis)

A-2010-2 W. Hämäläinen: Efficient search for statistically significant dependency rules in binary data. 163 pp. (Ph.D. Thesis)