U N I V E R S I T Y O F T A R T U
Faculty of Mathematics and Computer Science
Institute of Computer Science

Kristo Käärmann

Haplotype inference using overlapping segments

Master's Thesis

Supervisor: Jaak Vilo, *PhD*
Instructor: Sven Laur, *MSc*

TARTU 2006

Contents

List of Algorithms

Chapter 1

Introduction

1.1 Genetics background

Genetics research, aiming to understand the biological systems at molecular level, requires enormous amounts of data to be analysed in order to draw sound scientific conclusions. Modern biotechnology has empowered geneticists with high throughput tools, such as microarray based genotyping and expression profiling, to acquire data experimentally from living organisms. But not all of the needed data can easily be measured in experiments. In some cases computational methods have proven useful to get reliable molecular level information without explicit physical measurements. Computational techniques have proven useful for sequence alignment, gene finding, genome assembly, molecular 3D structure prediction, molecular interactions, and the modeling of evolution.

It has been estimated that 99.7% of human DNA is common in every person, and now that baseline sequence has been made publicly available, only the remaining 0.3% is interesting for genetic variation studies [Con05]. The DNA molecule, although actually broken up into chromosomes, can be represented by one long 3.4 billion letter string in a four letter alphabet, each letter representing one nucleotide. Usually genome studies are designed against the positions on the DNA-string, which are known to be polymorphic. The term *locus* is often used to refer to a position on the genome. A locus is said to be *polymorphic* when the DNA can vary in that position among different individuals of the same species. There have been about 3 million validated single nucleotide polymorphisms, or *SNP*-s, recorded in human genome—another 6 million SNP-s have been predicted and are yet to be verified.

Higher multicellular organisms, such as humans, posess two separate sets of chromosomes, which can be thought of as two separate genomes—one from either parent. The majority of human cells contain two genomes and are referred to as *diploid* cells. Reproductive cells, such as sperm and egg cells, contain just one copy of the genome and are called *haploid*. There are even species known to have triploid, tetraploid and higher polyploid cells, but we will be more interested in the case of diploid genomes. Note that each genome (DNA molecule) is made up of two complementary strands, but as they are complementary and carry no additional information they can be regarded as one genome and represented by

	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_8
g_1	AA	GT	GG	AT	AC	GT	GT	GG
g_2	AT	GG	CG	AT	CC	GT	GG	GG
g_3	AA	GT	CC	AA	CC	TT	GT	AG

Figure 1.1: A sample of 3 individuals with genotypes in 8 loci.

a single four-letter alphabet string. A diploid cell contains two DNA molecules, both of which contain two complementary strands. In humans and most animals, the two copies of genomes in diploid cells originate from two parent organisms and almost certainly differ at some positions. Therefore, a diploid genotype of a single nucleotide locus can be represented by a pair of aminoacids, for example *adenine-adenine* AA (both parents have passed on adenine) or *adenine-guanine* AG (one parent has passed on adenine, the other guanine) etc. In the case of different amino acids being represented in offspring's genomes, the locus is said to be *heterozygotic*; otherwise it is said to be *homozygotic*.

When considering a number of loci, genotyping techniques produce pairs of aminoacids observed in a given locus. For an example, when looking at 8 polymorphic loci in 3 humans, we would read out a matrix of 24 pairs as in Figure 1.1 from a microarray genotyping experiment.

For many research purposes this representation of genotypes is perfectly satisfactory, but there are cases when researchers are interested in the underlying haploid genomes or *haplotypes*. Haplotypes are needed when multiple mutations appear in the same gene product (protein) but also for studying recombination hotspots and understanding the (block) structure of the genome. Instead of looking at a sequence of pairs, it would be more useful to acquire pairs of independent haploid sequences. Genotype g_1 in Figure 1.1 contains 5 heterozygotic loci and therefore there are $2^{5-1} = 2^4$ valid pairs of haplotypes that could have produced this genotype. The table above gives just one possible haplotype reconstruction for the given genotype. The problem is to determine the true pair of haplotypes out of the many valid solutions.

Unfortunately, there is no easy or cheap method for determining haplotypes biochemically. In theory, one would need to isolate a large number of chromosomes originating from one parent to do that. Instead, there have been a number of methods developed to infer the most likely haplotypes from genotype data. Some of these techniques involve genotyping additional family members, but others rely

	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_8
g_1	AA	GT	GG	AT	AC	GT	GT	GG
$h_{1,1}$	A	T	G	T	C	G	T	G
$h_{1,2}$	A	G	G	A	A	T	G	G

Figure 1.2: An observed genotype and the underlying pair of haplotypes.

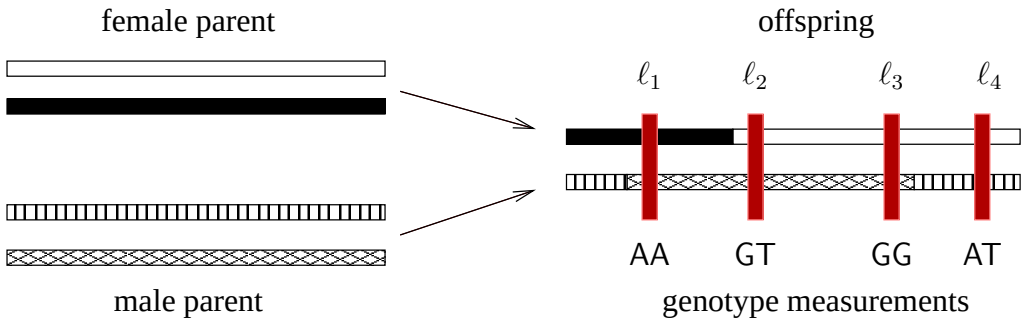


Figure 1.3: At meiosis stage, the two haploid DNA strings are combined to form a new haplotype and the offspring receives haploid DNA from both parents. Genotyping is locus specific and returns a pair of nucleotides. The measurement results at one locus are independent of other loci and it cannot be determined from which haplotype a particular nucleotide originated.

on statistical likelihood maximisation.

In the remainder of this work, we will be looking at the various computational methods for reconstructing the actual haplotypes for a population of individuals.

1.2 Motivation

The authors of this work were introduced with the haplotype inference problem during an earlier research project on learning haplotype block structure from HapMap datasets. These datasets are comprised of genome-wide diploid genotypes of about 300 individuals from 4 ethnic subpopulations [Con05]. To study the haplotype block structure, we first needed to acquire the haploid genome sequences of these populations. We used a popular haplotype inference tool [SSD01] (`phase`) for the purpose, but soon discovered the computational limitations, which did not allow to infer haplotypes of lengths that would be useful in the study of haplotype structure.

While being constrained by the computational complexity of haplotype inference, we started investigating techniques or methods, which would allow to acquire haplotypes from genotypes at a lesser computational expence, while possibly losing some accuracy across longer distances on sequences. We were eventually able to outline a framework that promises considerable time-efficiency and which will be described and analysed in the later chapters of this work.

As genotype acquiry methods have become cheaper and large datasets have been made available to reaserchers, there has been much active research in the field of haplotype inference. There is also a strong connection with population genetics and modelling that has made this task particularly interesting for various prominent research groups with statistics background. A variety of inference methods have been proposed over the last years from diverse approaches. A review of the known methods will be given in the next chapter of this work.

1.3 Contents of this work

Learning from research on existing hierarchical methods (described later) and borrowing from a *shotgun sequencing* technique, we came up with a parallelisable framework for haplotype inference. This setup uses existing inference tools at its core and uses preprocessing of input data along with postprocessing of results to constrain the search space. Our main contribution to the known hierarchical methods is taking advantage of sequence overlaps (as in shotgun sequencing) while partitioning a hard-to-solve long sequence into shorter segments. The easier-to-find haplotype reconstructions of the short overlapping segments are then ligated into the full sequence reconstruction according to the overlapping sequence, by heterozygous loci in particular.

To be able to measure the accuracy of our technique, we have put an effort into formulating a general framework to measure *haplotyping error* empirically using any inference method or software. We have also set up a number of biological and simulated datasets to obtain inference error rates with `phase` software.

Based on the empirical measurements, we constructed an *error model* and estimated characteristic parameters for `phase` on a set of semi-synthetic datasets. The error model enables us to broadly assess the error probabilities without running the software. The model is also an important input for understanding and estimating error propagation when applying our novel partition-ligation technique.

We then created wrapper scripts around `phase` to *implement our partition-ligation technique*. While doing this, we aimed at creating an optimised and high throughput solution. We made our wrapper software *grid-enabled* and tested it with `nordugrid` middleware in Estonian Grid.

Finally, we measured the empirical error rates with our technique and compared these to plain `phase` inference. We also constructed an error model for describing the probability of additional inference errors from our method.

This text has been structured as follows: first we explain existing haplotype inference methods along with genetics models that these methods employ. We then establish notation and universal models for studying empirical error rates in any haplotype inference process. We prepare test datasets from different origins: biological, re-processed biological and fully simulated ones. We are then set to propose our novel enhanced framework and are able to compare the performance and accuracy of our enhancements with the popular existing tools. As our method is far from being complete and final, we conclude this work with discussion and outlooks for further improvement.

Chapter 2

Relevant Biological Aspects

With this chapter we aim to give an introduction to the underlying biological aspects of haplotype inference. After introducing notation, we will mainly focus on the mathematical haploid population models. The treatment of the haplotype models pave way for discussing statistical haplotype inference methods in the following chapter.

2.1 Notation

We will now introduce basic notation that we are going to use in the following text. Let us use $G = \{g_1, \ldots, g_N\}$ to note a given population of genotypes. An input sample consists of genotype data for an ordered set of loci $\mathcal{S} = \{\ell_1, \ldots, \ell_k\}$ such that for every genotype and locus $g[\ell] \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}, \mathsf{H}, -\}$, where A,C,G,T mark homozygous genotypes of respective amino acids, H denotes a heterozygous locus and - marks unavailable data. We use $H = \{h_1, \ldots, h_m\}$ to refer to a set of haplotypes or sometimes to a haplotype reconstruction for G. In the latter case $m = 2N$. The fact that a genotype g is made up of two haplotypes h_1 and h_2 will be expressed as $g = h_1 \oplus h_2$. We also use H to denote the haplotype allele distribution in a population and respective allele frequencies by $\mathbf{p} = p_1, \ldots, p_m$. The meaning of H is given in context, weather it refers to a distribution, set or reconstruction.

We will expand our notation in the following chapters to accomodate partitioning and error models.

2.2 Biologically justified models

In many real world applications, researchers observe data being produced by some physical, biological, social or other processes. Often the model and the distribution, which produces the data sample is known and can formally be expressed in mathematical terms. This is the case with throwing the dice (uniform distribution), error in physical measurements (normal distribution), etc. In other cases the model is not explicitly known or cannot be fully described. Hypotheses and approximations can then be made about the model and the distribution, from which observed samples are drawn.

The latter case is true for many population genetics applications for it is usually impossible to describe the exact population structure, mating partner and environmental selection, geographical drift, etc. Furthermore, a level of uncertainty is still associated with understanding the underlying mechanics in the molecular level. The observed recombination rate variation[1] (crossing-over events are more likely to occur in some parts of the genome than elsewhere), for example, has not yet been sufficently explained by geneticists.

Due to the complexities and a great degree of unknowns concerning the biological systems, it has proven useful to make simplifying assumptions and approximations about the model that produces the data samples. In our case, by data samples we refer to the diploid genomes that can be observed in living organisms using modern genotyping techniques. We are also not necessarily interested in the full 3.4 billion base pair genome (for humans), but usually just a segment of it represented by a number of polymorphic loci (SNP's).

2.2.1 Probabilistic models of observed data

This subsection describes models that define the probability of observed data on the condition of presumed haplotype distribution, in short

$$\Pr\left[G|H\right].$$

Starting off with the search for a model to produce diploid genomes, it should be noted that the last step of the real world model is to pair up haploid genomes into diploid ones. Let it be a process or a model M_1

$$H \longmapsto_{M_1} G,$$

while $|H| = 2|G|$. For our original purpose of estimating the likelihood of H we need to describe the conditional probability with regards to the model that we are about to describe

$$\Pr\left[G|H, M_1\right] \quad .$$

The model used for M_1 usually assumes a simple random mating process without any selection. In statistical terms, one would refer to the model as sampling populations from a multinomial distribution of genotypes g, each with a probability $\Pr\left[g|H\right]$.

The only constraint by the model for calculating the genotype probabilities is that the haplotypes need to be valid for the genotype under consideration. By valid haplotypes, we refer to such haplotypes that have sequence properties for pairing up into the observed genotype. For example, haploid sequences GATGAG and GATGAC could never produce a diploid genotype GATHAG, because they conflict with the genotype in positions 4 and 6.

Let us use the notation

$$h_i \oplus h_j = g$$

[1]The crossing over events produce a novel haplotype at meiosis stage from diploid stem cells by combining segments from both sets of chromosomes. Jensen-Seaman *et al* [JSFP$^+$04] have reported an average 4-6 crossing over events per 10M base pairs per generation for humans. This totals to 1200-2000 recombinations for a production of one human haploid cell, e.g. sperm cell.

to express that a pair of haplotypes h_i and h_j would be valid to produce a genotype g. We can then write:

$$\Pr[g|H] = \Pr[(h_i, h_j) \mid h_i, h_j \leftarrow H,\ h_i \oplus h_j = g].$$

Let the haplotype frequencies in H be described by a probabilities vector $\mathbf{p} = p_1, \ldots, p_n$, then the above can be expanded further into

$$\Pr[g|H] = \sum_{h_i \oplus h_j = g} p_i p_j,$$

where we regard a genotype as an ordered pair of haplotypes. Both $h_i \oplus h_j$ and $h_j \oplus h_i$ produce a term in the sum. This conditional distribution formula is assuming that every individual has an equal chance of mating with any other individual in the population, which is the first serious (and completely unrealistic in real world) approximation of the model.

Having the model for describing how genotypes are obtained from haplotypes, let us proceed into formulating an approximate model for simulating sets (or populations) of realistic haplotypes. The statisticians would now be comfortable speaking about a *distribution of haplotypes*, from which one can sample haplotypes, each with a certain probability depending on distribution parameters. Recall that even if we are only allowing bi-allelic polymorphisms, there are 2^n potential haplotypes for a segment with n SNPs.

2.2.2 Ewens sampling process

In the two next chapters we will be concerned with a somewhat more difficult task, that is approximating the *prior* distribution of haplotypes $\Pr[H]$.

We will take a simplified population genetics model (namely Wright-Fisher model) and provide as much intuition as possible on how prior haplotype probabilities can be calculated out of such model. The US National Research Council has published a good introductive text on statistical models that are used in genetics [Cou05], which is suggested for further reading and formal derivations.

Ewens sampling formula is based on the Fisher-Wright neutral alleles model. This idealised model assumes, that the effective population from which the sample was drawn, is constant-sized, as every generation produces the same total number of offsprings to form the next generation. The non-overlapping generations are looked upon as discrete states of the population. According to the neutral model, no selection occurs (every allele is as competitive as any other) and the population has no group-structure, providing that any individual can mate and reproduce with any other individual in the population[2].

Assuming that every allele can equally produce any number of copies of itself (or none at all), one can view the creation of the next generation as sampling N individuals from an infinite pool according to allele frequencies. If there are only 2 alleles present in the population, this equals to sampling from a binomial

[2]The only difference from *panmictic* population is the constrained constant size of the populaton in Fisher-Wright.

distribution, in a k-alleles setting this generalises to sampling from a multinomial distribution.

It is interesting to note that if no mutations were allowed, eventually exactly one haplotype (a single sequence) would survive and take over the entire Fisher-Wright population. Every allele has a fixation probability equal to its original frequency. The phenomenon, also known as the *genetic drift*, can be verified by constructing a Markov process. Intuitively, the drift is due to the constant population size—in every generation there is a chance that an allele does not make it into the next generation of N individuals. The chance of dying out is higher for alleles with low frequency in the population.

For an addition to the purist model, one can allow to have novel haplotypes or mutations appear at a certain probability μ (or *mutation rate*) in every generation. We will consider now the *infinite alleles* mutation model, according to which a novel allele (i.e. new version of the sequence) is introduced to the population at the rate μ in every generation. The other approaches would be the k-alleles model, by which one of the k valid alleles would be (re-)introduced at the rate μ, and the infinite sites mutation model, which is a fine grained (base pair level) version of the infinite alleles model. It has also been shown that with mutations, the genetic variance asymptotically reaches a statutory distribution resulting from the stochastic balance between the mutations, which produce variation, and the genetic drift, which try to eliminate it.

Under such stochastic equilibrium, where mutations balance the genetic drift, the population model can be explained by a process also known as the *chinese restaurant process* [AIJ85] or sometimes also just an *urn model*. Let us imagine a chinese restaurant with potentially infinite number of tables, each capable of hosting infinite number of people. Each new customer either sits down at an already occupied table, with a probability proportional to the number of people already sitting at that table, or sits alone at a table not occupied, with a probability $\theta/(k+\theta)$, where k is the number of people already in the restaurant and θ is a constant parameter. Ewens sampling is an exactly similar process in terms of population and alleles – the $(k+1)$th sampled allele is either an existing one (proportional to the frequencies) or a new one (with an expected frequency θ). The probability of observing a new allele (or sitting down at an unoccupied table)

$$\frac{\theta}{k+\theta} \tag{2.1}$$

arises from viewing the sampling as two independent Poisson processes with rates k and θ. The probability that the first exponential event is of a particular type is just the relative rate of that event to the total rate.

The mutation parameter is obtained from

$$\theta = 4N\mu, \tag{2.2}$$

where N is the effective[3] diploid population size and μ is the mutation rate per generation. The trouble is, that normally one would not have the knowledge of

[3]effective population size can be taken as the harmonic mean of the population sizes over times

an average mutation rate nor the effective population size millions of years ago. Thus, the parameter θ is usually not calculated but estimated from genotype data.

Ewens gives a formula describing the structure of the population that has been obtained from such a model. It answers the questions, how many different alleles would we expect to observe in a finite population and what would be the distribution of allele frequencies. In the restaurant setting – how many tables would be occupied and how would the customers be distributed among tables after the Nth person has entered the restaurant. Let a_1 count the number of different alleles represented once in the population of N individuals, a_2 alleles being represented twice, etc. so that

$$a_1 + 2a_2 + \cdots + Na_N = N.$$

In the restaurant setting this notation suggests that there are a_1 tables occupied by just one customer, a_2 tables accomodating two and so forth. As $a_1, \ldots, a_N$ are integers, most of them have to be zero to satisfy this equality. A multivariate distribution also known as the *Ewens distribution* gives a probability for a Fisher-Wright model with a mutation parameter θ to produce a population that can be described by counts $a_1, \cdots, a_N$. The distribution can be computed using the formula

$$\Pr\left[a_1, \ldots, a_n | \theta\right] = \frac{\theta!}{\theta_{(n)}} \prod_{i=1}^{N} \left(\frac{\theta}{i}\right)^{a_i} \frac{1}{a_i!}, \tag{2.3}$$

where $\theta_{(n)}$ is the falling factorial. One can see that if the mutation parameter θ is close to zero, the probability reaches 1 for all the individuals carrying the same allele or, in a chinese restaurant, sitting at the same table. If $\theta = 1$, then all population structures have equal probability and taking $\theta \to \infty$ makes it highly likely that every individual carries a unique allele. It is also possible to derive the probability distribution for observing K_N different alleles $\Pr\left[K_N | N, \theta\right]$ in a population. Nevertheless, it is easy to notice from the setup of the process that the expected number of different alleles must be

$$\mathbf{E}(K_N) = \sum_{i=0}^{N-1} \frac{\theta}{\theta + i}.$$

Ewens sampling process is mirrored by the retrospective approach of the *coalescent theory* [Hud83, Nor02, Wak06]. The properties of the coalescent process have been shown to be supported by the Fischer-Wright model for large population sizes. Opposed to Ewens sampling, which describes how new alleles are being created as population evolves, the coalescent describes how the ancestral lineages of an observed sample coalesce when moving backwards in time.

Besides applications in genetics, the Ewens distribution has been used to describe models in ecology, particle physics and studying the spreading patterns of news and rumors. A thorough analyses of the properties of the distribution has been been provided by Tavare [ST98]. It has to be noted that Fischer-Wright model is in essence contradictory to the widely adopted Darwinian model, by which natural selection is the primary driver in development of populations.

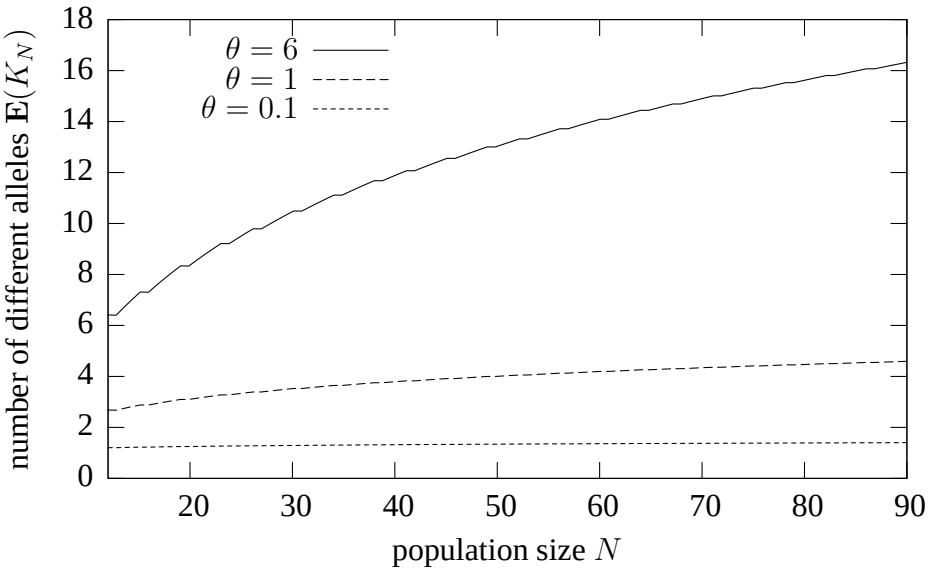


Figure 2.1: Number of different alleles in population grows according to parameter θ as the population size increases.

There have been numerous successful efforts and tools built for estimating the value of θ from an observed population of genotypes. Such applications allow to use more sophisticated models with some selection parameters, like population size bottlenecks, variable recombination rate, infinite sites (instead of infinite alleles) mutation model, and more. From these various enhancements to Ewens sampling, the models for haplotype inference have particularly benefited from enabling recombination and replacing the infinete alleles mutation model with a more specific infinite sites model [LS03].

2.2.3 Recombination and infinite sites mutation model

In nature, crossing over provides the mechanism for genetic recombination of haplotypes. To describe the crossing over formally, we need to differentiate between the *haplotype allele*, which refers to a distinct multi-locus sequence, and the *allele at a certain site* or locus on the haplotype region under examination, e.g. the allele of a SNP or a microsattelite[4]. In a genetic model, first a base haplotype allele is sampled from a population and a Markov process is let to iterate on the sequence of sites to model the effect of the crossing over. The process has then a potential probability p_j to *cross over to* another haplotype allele between sites ℓ_j and ℓ_{j+1}. The process is illustrated in Figure 2.2.

Li and Stephens [LS03] have built on the understanding that the crossing over events occur as a Poisson process along the sequence. Thus, the possibility of observing a crossing over at some point on sequence has exponential distribution

[4]Microsatellite is a repeated motif of nucleotides, usually only two or three bases in length. Microsatellites are used as genetic markers along with SNPs—a difference is that when SNP has 2 alleles, microsattelite usually has more

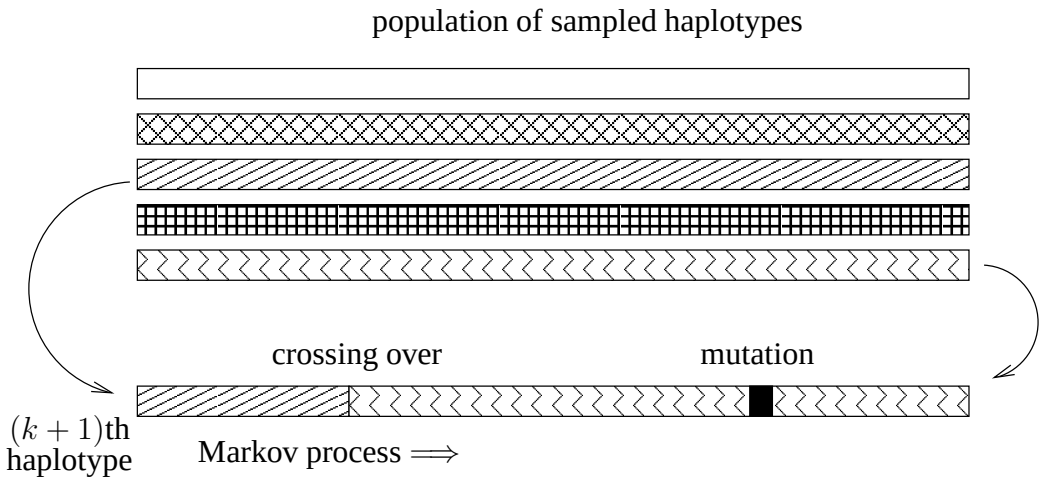


Figure 2.2: Illustration of a sampling process with recombination and infinite sites mutation model. Firstly, a haplotype is sampled from already sampled haplotypes. Then a Markov process iterates along the sequence and may cross over to another haplotype or insert a mutation.

with regard to the physical distance and recombination rate. Similar to modeling of mutations by Ewens, the site specific recombination parameter ρ_j depends on the effective (diploid) population size N and an assumed crossing over frequency c_j per physical unit distance between the sites ℓ_j and ℓ_j

$$\rho_j = 4Nc_j. \tag{2.4}$$

Denoting the distance weighted recombination parameter $r_j = d_j\rho_j$, where d_j is the distance in some units (such as base pairs), they have proposed formula 2.5 for calculating the probabilities in the transition matrix. The sampling in a recombination model has two steps: first a proposal haplotype is picked similar to Ewens sampling, and then a Markov process iterates on the sequence of the proposed haplotype – probably crossing over to some other already sampled haplotype. The probabilities for continuing ($x_{j+1} = x_j$) and crossing over ($x_{j+1} \neq x_j$) between loci j and $j+1$ is given by

$$\Pr\left[x_{j+1}|k,\mathbf{r}\right] = \begin{cases} e^{-r_j/k} + \frac{1-e^{-r_j/k}}{k} & \text{, if } x_{j+1} = x_j; \\ \frac{1-e^{-r_j/k}}{k} & \text{, if } x_{j+1} \neq x_j. \end{cases} \tag{2.5}$$

To mimic the effects of mutation the copying process may be imperfect. With probability $k/(\theta+k)$ the next site is an exact copy from the one, which was sampled from k haplotype alleles. There is then a probability $\theta/(\theta+k)$ that a mutation is applied to the site. While doing this, the effect of *back-mutation* must be taken into account—we can imagine a situation that a site mutates from $\mathsf{A} \rightarrow \mathsf{G}$ and after a number of generation mutates back to $\mathsf{G} \rightarrow \mathsf{A}$. Therefore, the actual probability for a mutation is cut by half, such that the probability of mutating a

site at the sampling process is

$$\Pr\left[\text{site mutation}\right] = \frac{\tilde{\theta}}{2(\tilde{\theta}+k)}$$

Taking N to be the number of chromosomes (haplotypes) in the sample, [LS03] have proposed to use

$$\tilde{\theta} = \left(\sum_{m=1}^{N-1} \frac{1}{m}\right)^{-1}$$

in place of θ, the motivation being described in [LS03].

The models described above have been used to assess the population parameters: mutation rate, effective population size, recombination rates and variation both from haplotype and genotype data [KYF00]. They also play an important role in haplotype inference. In particular for providing a prior distribution of haplotypes $\Pr\left[H\right]$.

2.3 Block structure in haplotypes

Empirical studies have confirmed that the human genome has a haplotype block structure [PBH$^+$01, Con05, ZCNS02], such that it can be divided into discrete blocks of limited haplotype diversity. This means that the *linkage disequilibrium* (LD) does not decay uniformly with distance as it has been shown by various empirical studies. Results from a recent large scale exploration of the fine-scale LD structure in humans has been given by [Con05]. There are regions, where the sequence is strongly linked together, separated by *recombination hotspots*, where it is highly likely that crossing over events occur. There have been suggestions that the hotspots are at least partly determined by some patterns on the sequence that facilitate the breaking of the DNA chain. Others argue, that the rarity of crossing over events during the short population history is a sufficient cause for diverse variation of recombination rates.

In each haplotype block, a small fraction of single-nucleotide polymorphisms, referred to as *tag SNPs*, can be used to distinguish a large fraction of the haplotypes. These tag SNPs can potentially be extremely useful for association studies, in that it may not be necessary to genotype all SNPs.

Eronen *et al* [EGT04] have used LD as a predictive factor for haplotype inference. They have trained variable order Markov models to capture linkage disequilibrium from sets of genotypes. Informally, the trained model describes the probability $\Pr\left[b|a,\ell\right]$ at some locus ℓ of observing a sequence b, if there was a sequence a preceding the locus. The trained Markov models are used to reconstruct haplotypes, choosing phase at each locus according to the probabilities $\Pr\left[b|a\right]$. The strength of LD, or the statistical correlation of loci, decays with distance due to recombination. Significant LD has generally been observed in the regions of 1-100 thousand base pairs. The estimated reach of LD hints the maximum reasonable backward-looking length of Markov models.

2.4 Synthetic simulation of haplotypes

In combination with population parameter estimation, the described statistical models provide a powerful tool for generating synthetic populations. When *in vitro* measurements only provide genotype data, the simulation methods are able to generate populations, for which the haplotype reconstructions are known. Available simulation implementations, such as [Hud02, MW06] rely on the Hudson coalescent process [Hud83] or its derivatives [MC05]. These tools sample coalescing geneaologies and insert mutations onto ancestral allele lineages. This is a different approach to Ewens sampling, but produces an equal outcome for large population sizes.

We have used some of these techniques in the generation of synthetic datasets in section 4.4.

Chapter 3

Haplotype Inference Methods

In this chapter we will review existing haplotype inference methods. We start with the simplest family approach known since the rise of Mendelian genetics. We also describe a simple heuristic approach but then contribute a more comprehensive section on statistical methods. The statistics section is set up to explain the general statistical inference methods and the setups for using the general methods in haplotype inference context. We describe the hierarchical approach to haplotype inference and give a preview into our proposed enhancements to haplotype inference.

3.1 Simple deduction methods

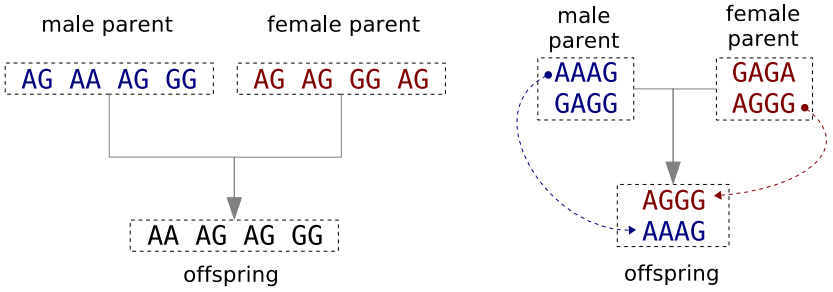
We will start the treatment of known inference methods by describing the most straight-forward appproaches accompanied by illustrating examples. Intuition gained from these simple methods allows us to advance towards more sophisticated statistical inference techniques in the following sections.

3.1.1 Direct deduction among family members

Using the direct deduction for haplotype inference, one would first need to acquire genotypes for full trios, each consisting of two parents and an offspring. Among parent-parent-offspring trios, one can use exclusion to determine individual haplotypes for both parents and an offspring, provided that at every locus at least one of them is homozygous.

Figure 3.1 demonstrates haplotype inference on a simple 4-locus case. Both parents have passed on A to the offspring at the first locus. As the male parent is homozygotic in the second locus, he must have passed on A and G must have originated from the female parent. Similarly, G must have come from the female parent at the third locus and the male could only have transmitted A. The offspring is homozygotic in the fourth position, which means that both parents have passed on G. Since the haplotype resolution is known for the offspring, we can solve the genotypes of both parents.

Although this technique is simple and fast, it still has several severe drawbacks. Firstly, it requires genotypes of full trios, which may be costly or some-



(a) Original genotypes as sequence of pairs

(b) Inferred haplotype reconstruction as pair of sequences

Figure 3.1: Haplotype inference with family method. The genotypes are solved locus-by-locus, excluding impossible combinations.

times impossible to obtain. Secondly, there is up to 25% probability that both parents and the offspring are heterozygotes at a given locus, in which case the family method cannot produce any solution. The exact probability of observing heterozygous locus is nevertheless dependent on allele frequency at that polymorphism.

3.1.2 Parsimony approach

Haplotype inference by parsimony was first introduced by Clark [Cla90] in 1990. Clark's greedy search algorithm starts off with a set of genotypes (a *population*) and at least one known haplotype. For example, an observed genotype that is homozygotic in every locus or has no more than one heterozygotic locus, presents one or two unambiguous haplotypes.

The algorithm first iterates through observed genotypes and identifies available unambiguous haplotypes in the population. It then searches genotypes that may have any of the existing haplotypes as one constituent. If it finds such a genotype, it derives a new haplotype by *subtracting* the original haplotype from the genotype. The subtraction procedure is illustrated in Figure 3.2.

The algorithm then continues with the updated set of haplotypes and iterates until no more genotypes can be solved. The problem with Clark's approach is that the algorithm might not even be able to start, unless there are already known haplotypes available. Also, it cannot be guaranteed that the method resolves all genotypes in the population. Even if it manages to solve all genotypes, the algorithm only provides one possible solution, which may or may not be the true reconstruction.

3.2 Statistical methods

Modern haplotype inference software tools today are built around statistical methods, such as Expectation-Maximisation (EM) [LM95] and Markov Chain Monte

	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_8
matching genotype	AA	GT	GG	AT	AC	GT	GT	GG
known haplotype	A	T	G	T	C	G	T	G
new derived haplotype	A	G	G	A	A	T	G	G

Figure 3.2: Deriving new haplotypes with parsimony approach. The process resembles *substracting* known haplotype from an observed genotype, thus creating a new haplotype.

Carlo simulations [SSD01]. A comprehensive overview of statistical methods and available tools has recently been published by Weale [Wea04].

In the introductury section we aim to give basic overview on how the statistical methods can be set up to infer the haplotype reconstruction from genotype data. We avoid formal derivation and proofs, but try to explain the intuition and give basic formulas that these methods rely upon.

When formulating the haplotype inference task in statistical notation, one needs to replace finding *the* haplotype reconstruction with finding *the most likely* haplotype reconstruction. Basis for the statistical haplotype inference is to acknowledge that many haplotype reconstructions are possible and some may even be equally likely with regards to information available to us at the time of inference.

3.2.1 Statistical estimation

Statistical estimation deals with estimating the values of hidden parameters based on measured data. The parameters usually describe some physical scenario that is unrevealed to the observer or only known to adhere to some model. In haplotype inference the parameters are the actual haplotypes, which we would like to estimate based on the observed genotypes.

Maximum likelihood estimation (MLE)

The principle of *maximum likelihood estimation*, originally developed by Fischer in the 1920s states that the desired probability distribution is the one that makes the observed data "most likely". This means that one must seek the value of the parameter vector (set of haplotypes H in our case) that maximises the probability of observed data (set of genotypes G).

From the previously described biological model for the distribution $\Pr[G|H]$ we may write

$$\mathcal{L}(H) = \Pr[G|H] \propto \prod_{g \in G} \Pr[g|H] \tag{3.1}$$

where $\mathcal{L}$ denotes a class of *likelihood functions* $f(G, H)$, whose value is proportional to $\Pr[G|H]$. In further discussion we will just use $\Pr[G|H]$ for the

likelihood function. Let $\mathbf{p}$ be the haplotype frequencies in H, then

$$\Pr[g|H] = \sum_{h_i \oplus h_j = g} p_i p_j$$

$$\hat{H}_{\texttt{MLE}} = \arg_H \max \prod_{g \in G}^{G} \sum_{h_i \oplus h_j = g} p_i p_j$$

Estimating the most likely set of haplotypes (or haplotype frequencies) boils down to finding a maximum on the likelihood surface. Since many probability density functions are exponential by nature, it may prove easier to maximise the *log likelihood* as the logarithm function is monotonous

$$\arg\max \mathcal{L}(H) = \arg\max \ln \mathcal{L}(H).$$

At the maximum point on surface $\mathcal{L}(\mathbf{p})$ the first derivative must vanish and to guarantee that it's a peak rather than valley, we need to check that the Hessian matrix is negative definite

$$\frac{\partial \mathcal{L}(H)}{\partial \mathbf{p}} = 0$$
$$\frac{\partial^2 \mathcal{L}(H)}{\partial^2 \mathbf{p}} < 0.$$

In practice it is usually difficult to calculate these derivatives analytically—especially in our case, where there are a huge number of parameters (as many as there are valid haplotypes for observed genotypes). Therefore, numerical methods have been used to maximise the likelihood or log-likelihood functions.

Maximum a posteriori (MAP)

The posterior mode estimation is very similar to the Fisher's MLE principle described earlier, with an added feature of taking the *prior* distribution for unobserved parameters into account. Let H denote a haplotype reconstruction and G be the observed set of diploid genotypes, then the probability of H being *the true* haplotype reconstruction is expressed as a *posterior probability*

$$\Pr[H|G]. \tag{3.2}$$

According to Bayes rule, the formula 3.2 can be expanded to

$$\Pr[H|G] = \frac{\Pr[G|H]\Pr[H]}{\Pr[G]}$$

by saying that the probability of H being *the* reconstruction depends on both the probability that such a reconstruction could produce the genotype sample we observed and also on the expected *prior* frequency of haplotypes in H. For an explanation of the importance of the prior, one might imagine a haplotype reconstruction that is valid for G, but is for some reason very unlikely to be observed

in nature (regardless of G). The normalising constant $\Pr[G]$ is not relevant for this task, as long as we assume that the observed genotypes were not selected specifically by any model. The above can also be described in terms of *likelihood*, saying that the posterior distribution of observed genotypes is proportional to the likelihood of H for a fixed G and the probability distribution of H, as in

$$\Pr[H|G] \propto \mathcal{L}(H) \Pr[H].$$

Thus, the maximum a posteriori parameter estimation can be expressed as

$$\hat{H}_{\texttt{MAP}} = \arg\max_{H}(\Pr[G|H] \Pr[H]).$$

Using the prior knowledge about the distribution of haplotypes in inference process has shown to increase the accuracy of the results in stochastic inference methods.

Following the Bayes notation, we now have two somewhat smaller tasks:

1. Firstly, calculate the probability, by which observed genotypes are produced from haplotypes H according to the biological model.

2. Secondly, calculate the probability of H according to the population models.

Having the models for computing $\Pr[G|H]$ and a distribution $\Pr[H]$, we need a method for maximising the compound probability.

3.2.2 Expectation-Maximisation (EM)

The *expectation-maximisation (EM) algorithm* first outlined by [Dem77] is a generalised numerical method for finding maximum likelihood estimation parameters in probabilistic models, where the model depends on unobserved *latent* variables (also known as hidden variables). EM alternates between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step, and the process is repeated.

General case

More formally, let $\mathbf{y}$ denote an observed sample of data produced by some random variable Y and $\mathbf{z}$ represent unobserved or missing data originating from Z. The unobserved data or *latent variables* together with observed data form the *complete data*. The aim is to draw inferences about a vector of parameters $\boldsymbol{\theta}$ with regard to the posterior density $p(\boldsymbol{\theta}|\mathbf{y})$, usually find the most likely $\boldsymbol{\theta}$. The use for EM arises when it is not possible or sufficiently accurate to calculate $p(\boldsymbol{\theta}|\mathbf{y})$ straight away, but achievable to calculate $p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z})$ if $\mathbf{z}$ where known.

The algorithm can be derived following

$$p\left(\boldsymbol{\theta}|\mathbf{y}\right) = \frac{p\left(\boldsymbol{\theta},\mathbf{z}|\mathbf{y}\right)}{p\left(\mathbf{z}|\boldsymbol{\theta},\mathbf{y}\right)},$$

taking logarithms from both sides leads to

$$\log p\left(\boldsymbol{\theta}|\mathbf{y}\right) = \log p\left(\boldsymbol{\theta},\mathbf{z}|\mathbf{y}\right) - \log p\left(\mathbf{z}|\boldsymbol{\theta},\mathbf{y}\right)$$

and as the left hand side does not rely on $\mathbf{z}$, we can sum the right hand side over the conditional distribution of $[\mathbf{z}|\boldsymbol{\theta}_t,\mathbf{y}]$, as

$$\log p\left(\boldsymbol{\theta}|\mathbf{y}\right) = \int p\left(\mathbf{z}|\boldsymbol{\theta}_t,\mathbf{y}\right)\log p\left(\boldsymbol{\theta},\mathbf{z}|\mathbf{y}\right)d\mathbf{z} - \int p\left(\mathbf{z}|\boldsymbol{\theta}_t,\mathbf{y}\right)\log p\left(\mathbf{z}|\boldsymbol{\theta},\mathbf{y}\right)\,d\mathbf{z}$$

It has been shown that in order to maximise the probability of $\boldsymbol{\theta}$, one can discard the second term on the right hand side and maximise only the first term often referred to as $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t)$. Using the derivation, we are able to maximise $p\left(\boldsymbol{\theta}|\mathbf{y}\right)$, provided that we can estimate missing data $p\left(\mathbf{z}|\boldsymbol{\theta}_t,\mathbf{y}\right)$ and maximise the parameters from complete data $p\left(\boldsymbol{\theta},\mathbf{z}|\mathbf{y}\right)$. The EM algorithm thus iteratively maximises the function

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t) = \int p\left(\mathbf{z}|\boldsymbol{\theta}_t,\mathbf{y}\right)\log p\left(\boldsymbol{\theta},\mathbf{z}|\mathbf{y}\right)d\mathbf{z}. \tag{3.3}$$

In the **expectation step** (E), one calculates the integral as a function of $\boldsymbol{\theta}$, given the estimate $\boldsymbol{\theta}_t$ from the previous step. In the **maximisation step** (M), the new estimate is taken at the maximum of the calculated function $\boldsymbol{\theta}_{t+1} = \arg\max Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t)$. The maximisation can be done either analytically or heuristically, depending on the actual function.

Haplotype inference using EM

The EM framework for estimating the haplotype configuration from genotypes was proposed by [LM95], but more clearly defined in [NQXL02]. Here the unknown data is the haplotype configuration $\mathbf{z} \leftarrow Z \subset H \times H$ consisting of pairs of haplotypes, which all together equal to the observed genotypes[1] G. In the general EM notation genotypes G corresponds to the observed data Y, haplotype resolution H is the latent variable and the vector of haplotype frequencies corresponds to parameters $\boldsymbol{\theta}$. Therefore, the EM process for haplotype inference is set up to maximise the likelihood of haplotype frequencies $\mathbf{p}$ the corresponding maximum likelihood haplotype resolution H is just a byproduct. Note that we are not able to calculate $p\left(\mathbf{p}|G\right)$ directly, but we can compute $\Pr\left[H|G,\mathbf{p}\right]$ and $p\left(\mathbf{p},H|G\right)$.

The expectation step requires explicit formula for $\Pr\left[H|\boldsymbol{p},G\right]$. The latter is straightforward, as in previous section we derived

$$\Pr\left[h_i \oplus h_j = g|\boldsymbol{p}\right] = \frac{p_i p_j}{\sum\limits_{h_x \oplus h_y = g} p_x p_y}$$

[1] note that the haplotype confi guration uniquely determines G, as $Z \Rightarrow G$.

and by our assumptions all haplotype pairs are sampled independently. Next, we have to express $\Pr[\boldsymbol{p}, H|G]$. Due to the Bayes formula

$$p(\boldsymbol{p}, H|G) = \frac{\Pr[G|\boldsymbol{p}, H] \cdot \Pr[H|\boldsymbol{p}] \cdot p(\boldsymbol{p})}{\Pr[G]} = \prod_{g \in G} \frac{\Pr[g|\boldsymbol{p}, H] \cdot \Pr[H|\boldsymbol{p}] \cdot p(\boldsymbol{p})}{\Pr[g]} \ . \tag{3.4}$$

As haplotype resolution uniquely determines genotype, hence for consistent haplotype resolutions $H = (h_{1_1} \oplus h_{2_1}, \ldots, h_{1_n} \oplus h_{2_n})$,

$$p(\boldsymbol{p}, H|G) = \prod_{k=1}^{n} \frac{\Pr[h_{1_k} \oplus h_{2_k}|\boldsymbol{p}] \cdot p(\boldsymbol{p})}{\int_{\boldsymbol{p}} \Pr[g_k|\boldsymbol{p}]\, p(\boldsymbol{p})\, d\boldsymbol{p}} = \prod_{k=1}^{n} \frac{\Pr[h_{1_k} \oplus h_{2_k}|\boldsymbol{p}]}{\int_{\boldsymbol{p}} \Pr[g_k|\boldsymbol{p}]\, d\boldsymbol{p}}$$

if one assumes constant non-informative prior $p(\boldsymbol{p}) = 1$ for $\boldsymbol{p} \in [0,1]^n$. In other words, a priori no parameters are favored. Note that the denominator is independent from $\boldsymbol{p}$ and thus

$$p(\boldsymbol{p}, H|G) \propto \prod_{k=1}^{n} p_{1_k} p_{2_k}$$

where the constant is independent from $\boldsymbol{p}$. Hence, the corresponding $Q(\boldsymbol{p}|\boldsymbol{p}_t)$ is equal up to additive constant

$$\begin{aligned} Q_0(\boldsymbol{p}|\boldsymbol{p}_t) &= \sum_{H \Rightarrow G} \Pr[H|\boldsymbol{p}_t, G] \sum_{k=1}^{n} (\log p_{1_k} + \log p_{2_k}) \\ &= \sum_{H \Rightarrow G} \sum_{k=1}^{n} \Pr[h_{1_k} \oplus h_{1_k} = g_k|\boldsymbol{p_t}] \left(\log p_{1_k} + \log p_{2_k}\right) \ . \end{aligned}$$

In the maximisation step we need to maximize $Q(\boldsymbol{p}|\boldsymbol{p}_t)$ w.r.t. constraint $p_1 + \cdots + p_m = 1$ where m is the possible number of haplotypes. The method of Lagrange' multipliers lead to a functional

$$Q^* = Q_0(\boldsymbol{p}|\boldsymbol{p}_t) + \lambda(p_1 + \cdots + p_m - 1) \ . \tag{3.5}$$

The corresponding partial derivatives are

$$\begin{aligned} \frac{\partial Q^*}{\partial p_a} &= \sum_{H \Rightarrow G} \sum_{k=1}^{n} \Pr[h_{1_k} \oplus h_{1_k} = g_k|\boldsymbol{p_t}] \cdot \frac{\partial}{\partial p_a} (\log p_{1_k} + \log p_{2_k}) + \lambda \\ &= \sum_{H \Rightarrow G} \sum_{k=1}^{n} \Pr[h_{1_k} \oplus h_{1_k} = g_k|\boldsymbol{p_t}] \left(\frac{\delta_{1_k=a}}{p_{1_k}} + \frac{\delta_{2_k=a}}{p_{2_k}} \right) + \lambda \\ &= \frac{1}{p_a} \cdot \mathbf{E}(n_a|\boldsymbol{p_t}, G) + \lambda \end{aligned}$$

where $\delta_{i=j}$ is a Kroneker symbol and $\mathbf{E}(n_a|\boldsymbol{p_t}, G)$ is the expected number haplotypes h_a in G w.r.t. haplotype frequencies $\boldsymbol{p_t}$. Now, taking $\frac{\partial Q^*}{\partial p_a} = 0$, we get

that $p_a \propto \mathbf{E}(n_a|\boldsymbol{p_t}, G)$. Since $\mathbf{E}(n_1|\boldsymbol{p_t}, G) + \cdots + \mathbf{E}(n_m|\boldsymbol{p_t}, G) = 2|G|$, we have derived the maximisation step

$$p_a = \frac{\mathbf{E}(n_a|\boldsymbol{p}_t, G)}{2|G|} \quad .$$

Hence, we have derived the EM-algorithm. For calculating the maximisation step, we follow

$$\mathbf{E}(n_a|G, \mathbf{p}_t) = \sum_{g \in G} \frac{\Pr\left[h_a \oplus h_x = g | \forall x\right]}{\Pr\left[g|H\right]}$$

$$\Pr\left[h_a \oplus h_x = g | \forall x\right] = \sum_{x : h_a \oplus h_x = g} p_a p_x$$

$$\mathbf{E}(n_a|G, \mathbf{p}_t) = \sum_{g \in G} \frac{\sum\limits_{x : h_a \oplus h_x = g} p_a p_x}{\sum\limits_{x,y : h_x \oplus h_y = g} p_x p_y}$$

Haplotype frequencies in $\mathbf{p}$ are updated according to the calculated estimation

$$p_a^{t+1} = \frac{\mathbf{E}(n_a|G, \mathbf{p}^t)}{2|G|} \tag{3.6}$$

The EM algorithm, described in Algorithm 1 iterates upon (3.6) until the expectation (3.5) does not change more than some very small predefined constant. It thus delivers the most likely haplotype frequency distribution $\hat{\mathbf{p}}$, from which it would be really straightforward to sample the *hidden* variables—haplotype reconstruction Z.

The intuition behind these formulas says that haplotypes that could be part of many observed genotypes are more likely than those that could be used to construct only few. The maximisation process thus leans towards minimising the number of different haplotypes and seeking in some natural sense the *optimal* set of underlying haplotypes. Once we have obtained the most likely haplotype frequencies in the population, we only need to draw two valid haplotypes for every $g \in G$ according to these frequencies.

Practical implementations of EM are only usable for small samples and short sequences, as the parameter space (number of valid haplotypes) is exponential to the highest number of heterozygote sites in any of the sample genotypes. Specifically, to calculate the expected haplotype counts, one needs to contain the probabilities of all possible haplotypes (which are valid for G) in memory and sum over these at every expectation step. Just for a reminder, there are potentially 2^k different valid haplotypes when the sequence contains k SNPs.

3.2.3 Markov Chain Monte Carlo (MCMC)

Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that

Algorithm 1: Expectation-Maximisation algorithm

Input: Observed data G. The set of all possible haplotypes H that could be valid for G.
Output: Haplotype frequencies $\hat{\mathbf{p}} = \arg\max \Pr[\mathbf{p}|G]$

function EM(G)
Initialise $\mathbf{p}^0$, so that $p_i^0 = 1/|\mathbf{p}|$.
repeat
  foreach $g \in G$
    initialise $c = 0$ for estimating $\Pr[g|\mathbf{p}]$.
    foreach $(h_i, h_j) \in H \times H$
      if $h_i \oplus h_j = g$ **then**
        $c = c + p_i^{(t)} p_j^{(t)}$
        $p_i^{(t+1)} = p_i^{(t+1)} + p_i^{(t)} p_j^{(t)}$
        $p_j^{(t+1)} = p_j^{(t+1)} + p_i^{(t)} p_j^{(t)}$
      end if
    end for
    $\mathbf{p}^{(t+1)} = \frac{\mathbf{p}^{(t+1)}}{c}$
  end for
  $\mathbf{p}^{(t+1)} = \frac{\mathbf{p}^{(t+1)}}{2|G|}$
until $Q(\mathbf{p}|\mathbf{p}^{(t+1)}) \leq Q(\mathbf{p}|\mathbf{p}^{(t)}) + \epsilon$
return $\mathbf{p}^{t+1}$.
end function

Algorithm 2: Metropolis-Hastings algorithm

Input: Observed data $\mathbf{y}$, number of iterations n.
Output: Sample from approximated distribution $\pi(\mathbf{x}|\mathbf{y})$

function METROPOLIS-HASTINGS(n)
Initialise $\mathbf{x}^0$.
for $t = 0$ **to** $n - 1$
  sample $\mathbf{x}^\star \leftarrow q(\mathbf{x}^\star|\mathbf{x}^t)$.
  sample $u \leftarrow \mathrm{U}(0, 1)$.
  if $u \leq a(\mathbf{x}^t, \mathbf{x}^\star)$ **then** $\mathbf{x}^{t+1} = \mathbf{x}^\star$
    else $\mathbf{x}^{t+1} = \mathbf{x}^t$
end for
return $\mathbf{x}^{t+1}$.
end function

has the desired distribution as its stationary distribution. The state of the chain after a large number of steps is then used as a sample from the desired distribution. The quality of the sample improves as a function of the number of steps. The classic problem for MCMC is solving of high-dimensional definite integrals, but there are popular applications for optimisation and model (or model parameters) inference. A comprehensive treatment of MCMC algortihms with several example applications in genetics can be seen in [DS02].

A popular MCMC algorithm is the Metropolis-Hastings algorithm, which allows to draw samples from any probability distribution $\pi(x)$, requiring only that the density can be calculated at x. Since direct sampling from the desired distribution $\pi(x)$ may be not be possible, the Metropolis-Hastings algorithm starts by generating candidate draws from the so-called *proposal distribution*. These draws are then corrected so that they behave, asymptotically, as random observations from the desired equilibrium of Markov chain (target distribution π). The Markov chain produced by the algorithm is thus constructed at each stage in two steps: a proposal step and an acceptance step. Let $\pi(\mathbf{x})$ be the desired distribution and $\mathbf{x}^t$ denote the state of the Markov chain at a given stage t. The next state of the chain is chosen by first sampling a candidate point $\mathbf{x}^\star$ from a proposal distribution $q(\mathbf{x}^\star|\mathbf{x}^t)$. The candidate point is accepted with a probability $\min\{a(\mathbf{x}^t, \mathbf{x}^\star), 1\}$, where

$$a(\mathbf{x}^t, \mathbf{x}^\star) = \frac{\pi(\mathbf{x}^\star)}{\pi(\mathbf{x}^t)} \frac{q(\mathbf{x}^t, \mathbf{x}^\star)}{q(\mathbf{x}^\star, \mathbf{x}^t)}.$$

If the candidate is accepted, it is taken as the new state of Markov chain $\mathbf{x}^{t+1} = \mathbf{x}^\star$, otherwise the state will not change and $\mathbf{x}^{t+1} = \mathbf{x}^t$.

The better $q(\mathbf{x}|\mathbf{x}^t)$ resembles $\pi(\mathbf{x})$, the faster the Markov chain converges to its equilibrium distribution, thus the art of making the algorithm work is choosing a good proposal distribution $q(\mathbf{x}|\mathbf{x}^t)$. The Metropolis-Hastings algorithm can be seen as a special case of the *simulated annealing* method [KGV83]. In simulated annealing, the sampling distribution $q(\mathbf{x}|\mathbf{x}^t)$ relies on a *temperature* parameter that gradually decreases over iterations. Higher temperature causes more random

Algorithm 3: Gibbs sampling algorithm.

Input: Observed data $\mathbf{y}$, number of iterations n.
Output: Sample from approximated distribution $\pi(\mathbf{x}|\mathbf{y})$, $|\mathbf{x}| = d$

function GIBBS($\mathbf{y}, n$)
Initialise $\mathbf{x}^0$.
for $t = 0$ **to** $n - 1$
  Pick random coordinate $j : 1 \leq j \leq d$.
  Update coordinate by sampling $x_j \leftarrow \pi(x_j|\mathbf{y}, \mathbf{x}_{-j})$.
end for
return $\mathbf{x}$.
end function

draws, while at low temperatures the samples conform to the expected distribution.

Gibbs sampler is an MCMC algorithm, a special case of Metropolis-Hastings, which is particularly popular due to its computational simplicity. Gibbs sampling produces a reversible Markov chain, in which new values are accepted at every stage. The trick to achieve this property, is to sample single coordinates from a conditional proposal distribution, assuming that the other coordinates already have the desired stationary distribution. Let $\mathbf{x} = (x_1, \ldots, x_d)$, then a coordinate i is sampled from a conditional distribution

$$\pi(x_i|\mathbf{x}^t_{-i}, \mathbf{y})$$

where $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, \cdot, x_{i+1}, \ldots, x_d)$, observed data denoted by $\mathbf{y}$ and the desired distribution $\pi(\mathbf{x})$.

The only art in implementing the Gibbs sampler is to be able to draw one-dimensional coordinate samples from a conditional distribution $\pi(x_j|\mathbf{y}, \mathbf{x})$.

Gibbs sampler for haplotype inference

Gibbs sampler constructed by Stephens *et al* [SSD01] has been implemented in a popular haplotype inference tool `phase`. An earlier article [SD00] provides thorough background and study of models with complete derivation of this algorithm.

The `phase` algortithm in its basic form starts off from an arbitrary haplotype configuration $\{h_1 \ldots, h_{2m}\} = H^{(0)}$ for the observed set of genotypes G, $|G| = m$. In every iteration t it draws a random individual g and samples a replacement pair of haplotypes $\{h_u, h_v\}^t$ for g from

$$\{h_u, h_v\}^t \leftarrow \Pr\left[H_i|G, H_{-i}\right].$$

Here H_g refers to the set of valid haplotype configurations for $g \in G$ and H_{-g} denotes the current haplotype reconstruction, but with haplotypes chosen for g being removed. In this way the algorithm iterates through reconstructions (Markov states) $H^{(0)}, H^{(1)}, H^{(2)}, \ldots$ and eventually reaches a stationary distribution $\Pr\left[H|G\right]$.

Unfortunately the conditional distribution $\Pr[H_g|G, H_{-g}]$ (or any $\Pr[H]$ for that matter) is not known for most realistic population genomics models. Specifically the following simplification is applicable to most coalescent based models.

$$\Pr[H_i|G, H_{-g}] \sim \Pr[H_g|H_{-g}] \sim \Pr[h_u|H] \Pr[h_v|H, h_u]$$

The haplotypes are sampled from a distribution where a probability of drawing a haplotype h is proposed to be given by

$$\Pr[h|H_g] = \frac{r_h + \theta p_h}{r + \theta}.$$

Here r refers to the count of different haplotypes in the obtained reconstruction, r_h is the count of h, θ is the mutation rate and p_h is the probability of h. This formula arises similarly to the Ewens sampling described earlier with added possibility that another allele may also mutate into h. The first simplifaction to the formula is to replace the unknown frequencies p_h with $1/M$, where M is the number of all different haplotypes that are valid to appear in any reconstruction of G. It is still hard to calculate the probabilities for every h, because H_i can be extremely large – if there are k heterozygote SNP-s in g_i then the number of possible haplotypes is $|H_g| = 2^{k-1}$. It was thus worth to notice that the probabilities $\Pr[h|H_g]$ are the same for every haplotype, which is not present in the currently constructed set (for which $r_h = 0$).

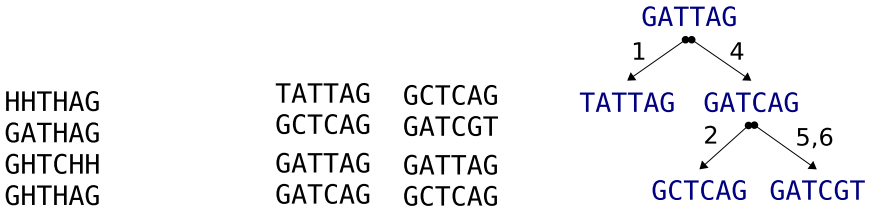
The accuracy of this naive version of Gibbs sampler is comparable to that of EM, but it is computationally much more feasible when the input genotype sequences are longer than 5-10 SNPs. The simpler version of `phase` algorithm is reproduced here in Algorithm 4. It is worth noticing, that the algorithm takes mutation rate θ as an input parameter. There are several MCMC algorithms, which allow to estimate θ from an observed population of genotypes. The `phase` software readily contains the algorithms that first estimate the population parameters to be used in the Gibbs sampling phase.

The actual version of the algorithm implemented in `phase` software is based on a more sophisticated and accurate model, by which new haplotypes are not created by randomly choosing phase at every heterozygous locus, but by applying s mutations drawn from geometric distribution to a haplotype already existing in the reconstructed set. This is an infinite sites mutation model, which also allows recombination.

3.3 Phylogeny method

Alternative way for inferring the most likely reconstruction of haplotypes is to resolve the history of mutations in the population. The evolutionary history of a haplotype can be expressed in a tree-like format, such a tree would be called a *phylogenetic tree*. A set of haplotypes (e.g. a haplotype reconstruction for observed genotypes) realises perfect phylogeny model if it is possible to construct a corresponding single-rooted phylogenetic tree with the following properties:

1. every vertex of the tree represents one observed haplotype;



(a) Observed genotypes

(b) A proposed haplotype reconstruction

(c) Perfect phylogeny tree corresponding to the reconstruction

Figure 3.3: Haplotype inference with phylogeny method attempts to seek haplotype reconstruction that is both valid for observed genotypes and satisfies *perfect phylogeny* conditions

2. if there is an observed polymorphism at site i, then there exists exactly one parent-child pair of nodes, so that the parent and the child represent different alleles of the site;

3. every haplotype of the original set is present in the phylogeny tree.

The perfect phylogeny assumes infinite sites mutation model, where there is no *back mutation*. In its purist form the phylogeny model also denies recombination. An example of a perfect phylogeny tree for a small set of haplotypes is given in Figure 3.3. Haplotype inference using phylogeny assumes, that the true haplotypes for some set of observed genotypes also carry the full history of haplotypes up to the most recent common ancestor. Meaning that when new mutant alleles appeared in the population, the original haplotypes also survived and are now represented in the observed set of individuals. Having only limited number of genotypes available, it is quite likely that some linking haplotypes in the historical tree are not present in the observed populations. It would also be reasonable to include back mutation and recombination in the model. Therefore, in more practical results *imperfect phylogeny* is formulated, see [EHK03] for further reference.

Almost linear-time algorithms for deriving a perfect haplotype phylogeny tree from genotypes have been described by Gusfield [Gus02] and Eskin *et al* [EHK03]. Adjustments to accomodate imperfect phylogeny models have also been proposed. The known fast algorithms either construct one valid phylogeny tree or return with verification that no such tree can be built on given data. If longer sequences are considered for input, it is usually the case that more than one valid phylogeny tree exist. For such cases, the algorithms have been adapted to return more than one possible solution. These different valid phylogeny reconstructions are then used as input for likelihood maximisation, using EM or other methods. Likelihood maximisation is now more feasible, because not all reconstructions are considered, but only the ones that adhere to perfect phylogeny.

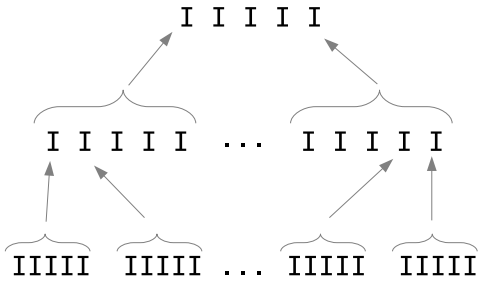


Figure 3.4: Schematics of partition-ligation (PL) technique. Initial sequences are broken up into segments, which are ligated hierarchically. In ligation phases, the segments are taken as multi-allelic sites, with inferred haplotypes as their possible alleles.

3.4 Partition-Ligation technique

As the search space for statistical algorithms is exponentially dependent on the number of heterozygous sites in the sample, the naive implementations cannot cope well with sequence lengths exceeding a few hundred sites. Niu [NQXL02] proposed a technique for constraining the search space, which has brought practical benefits to many software implementations. The general idea is to infer correct haplotype phase locally in segments and then hierarchically ligate the whole sequence together.

At first, the partition-ligation (PL) algorithm divides the given genotypes into ca. 10 site (SNP) segments and infers correct phase in each segment separately using `phase` or EM.

It then regards the segments as multiallelic sites, for which the alleles are defined by segment haplotypes, and creates a new partitioning, where every new segment contains about 10 original segments. The correct phase is then inferred for this new partitioning. The process continues until the full sequence has been ligated.

Any inference algorithm that supports phase detection on multiallelic sites, can be used with partition-ligation technique. It has been reported that the accuracy of haplotype reconstruction is similar or even better when one uses a PL wrapper around the original algorithm.

By constraining the number of sites to a constant $k \approx 10$, the original 2^{h-1} search space (where h is the number of heterozygotic sites) is now constrained with $h \leq k \approx 10$. On the other hand, due to the multiallele effect, the space has now the maximum size a^{k-1}, where a is the number of haplotypes for any segment.

Let us examine an inference task for a sequence of 300 biallelic polymorphisms, genotyped on a population of 100 individuals. Assuming the general heterozygosity 20%, the original search space would be $100 * 2^{60} \approx 10^{20}$ reconstructions. Using the PL technique, we have 30 separate 2^6 search spaces to start with. In the last ligation step, however, we would need to ligate 10 multiallelic 'sites', all of which actually contain 30 polymorphisms. We can expect the number of alleles in such artificial sites to be close to 200—every haplotype would

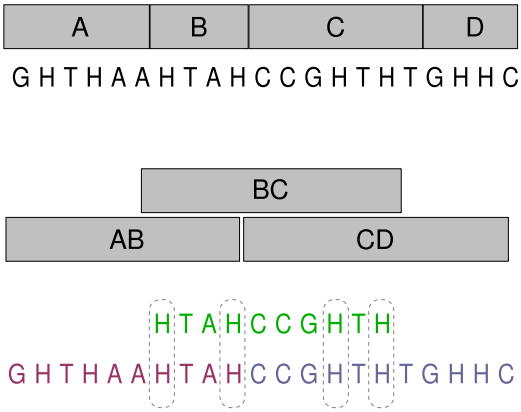


Figure 3.5: Ligation with segment overlapping. The sequence is partitioned into overlapping segments so, that the overlaps contain heterozygotic SNPs for as many genotypes as possible.

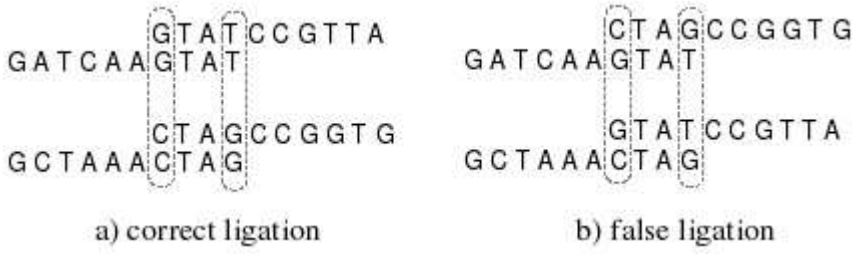


Figure 3.6: Determining correct ligation phase. The phase of heterozygotic loci in the overlapping region determine the correct way to ligate these together.

be unique for a 30 SNP stretch. Therefore, the size of search space for the latest phase can already be $200^{10} \approx 10^{23}$.

3.5 Overview of Partition-Ligation with segment overlapping

The proposed ligation method takes advantages of overlapped segments and heterozygos SNPs in the overlaps to infer correct phase for segment ligation. The partition ligation by overlapping can generally be described by the following steps:

1. Partition given SNP sequence into overlapping segments so that overlaps contain heterozygotic SNPs (see Figure 3.5). The partitioning aims to minimise the length of overlaps, while keeping the number of heterozygous overlaps above a given threshold (e.g. 80% of all overlaps).

2. Infer haplotypes for segments using `phase`.

3. Ligate all those segments that have sufficient number of heterozygotic SNPs in the overlap (see figure 3.6).

4. Regard ligation as a probabilistic event and calculate ligation confidence.

5. Infer ligation phase with likelihood methods for those neighbouring segments that are homozygous in the overlap. The ligations determined by overlapping can be used as known data.

There are a number of questions needed to be answered before finding the optimised solution. We need to understand what is the reasonable size for the segments and how to determine the *goodness* of a partitioning. For ligation, we need a probabilistic model for calculating the probability or *confidence* for the ligation. Once the possible ligations have been applied, we need to assess the accuracy of ligations on test data, whereas measuring and quantifying the accuracy of haplotype inference is an important topic by itself. To assess the prior error, we need to understand and model the error characteristics of `phase`, which is our core inference tool.

In the next chapter we will build up a framework for empirically assessing the accuracy of haplotype inference methods. This includes also specifying error models and establishing datasets for testing the methods.

Algorithm 4: Gibbs sampling algorithm for haplotype inference.

Input: Observed genotypes G, number of iterations n, scaled mutation rate θ.

Output: Sample drawn from approximated distribution of haplotype reconstructions $\Pr[H|G]$.

function PHASE(G, θ, n)
Initialise H with a random valid haplotype reconstruction
for $t = 0$ **to** $n - 1$
  Pick individual $g \in G$ at random
  foreach $h_i \in H$
    if h_i is a valid constituent for g **then**
      if $\exists h_j \in H$, such that $h_j \oplus h_i = g$ **then**
        $p_i = \frac{r_i + \theta/M}{r_j + \theta/M} - (\theta/M)^2$
      else
        $p_i = \frac{r_i \theta}{M}$
    else
      $p_i = 0$
  end if
  Let k be the number of valid reconstructions for g, $k = |H_g|$
  Sample u from uniform distribution $u \leftarrow U(0, 1)$
  if $u < \frac{2^k(\theta/M)^2}{\sum_j p_j + 2^k(\theta/M)^2}$ **then**
    Reconstruct g completely at random
  else
    Sample valid haplotype $h_j \leftarrow H_g$ according to $\frac{p_j}{\sum_{i \in H_g} \mathbf{p_i}}$
    Construct a counterpart $h_\star$, such that $h_j \oplus h_\star = g$
  end if
  Update H by setting $H = H_{-g} \cup \{h_j, h_\star\}$
end for
return H
end function

Chapter 4

Measuring Haplotype Inference Accuracy

The main purpose of this chapter is to set up a framework for measuring the accuracy of haplotype inference process. Such setup should be applicable to all different haplotype inference methods. Therefore, we focus on the empirical measurement opposed to theoretical analysis. Still, the knowledge about existing inference methods and population genetics, presented in Chapters 2 and 3, provides neccessary insight. It allows us to faithfully model the error situations and propose some universal stable statistics for quality assessment. The latter allows us to compare various inference methods and assess accuracy of our enhanced partition-ligation technique.

Firstly, we revisit basic notation and define error scenarios under a couple of different assumptions. We then prepare three datasets, for which we know the true haplotype reconstruction. Using `phase` inference tool, we obtain empiric measurements, and study error rates under various special conditions. As a result, we get approximate distribution of error rates together with their variance.

4.1 Notation

Recall that a genotype g consists of two chromosome alleles h_1 and h_2, briefly denoted $g = h_1 \oplus h_2$ while the order of maternal and paternal alleles is irrelevant. As each DNA strand consists of different four base-pairs A,C,G and T, we can represent a chromosome allele as a DNA string over a four letter alphabet A,C,G,T. The position of a base-pair is called a locus, i.e. the ith position of h_1 and h_2 correspond to the ith locus on the DNA. However, as the DNA is redundant, most of the loci contain exactly the same base-pairs in every individual and therefore usually only polymorphic loci (SNP-s) are considered.

More formally, let N be the whole length of the genome and let an observed sample contain a set of polymorphic loci $\mathcal{S} \subseteq \{1, \ldots, N\}$, i.e. for each $i \in \mathcal{S}$ there is a SNP at the ith locus of the genome. Then we can define the *condensed representation* of a genotype with regard to our sample as $g[\mathcal{S}] = h_1[\mathcal{S}] \oplus h_2[\mathcal{S}]$ where $[\cdot]$ denotes the subscript selection. In the following, we use only condensed representations and omit subscript selector, i.e. g is an ordered list of SNP values.

In principle, there are four possible SNP values, however, for small samples that are used currently in genetic studies rarely more than two values occur (biallelic SNPs). In genetics literature, the SNP allele with highest frequency is called a *wild type* and the less frequent is referred to as a *mutant allele*. We will avoid these terms, because there is no need for us to assume that one allele has considerably higher frequency over the other. We have earlier introduced H to denote heterozygous loci, the new notation is more specific but $g[\ell] = \mathsf{AC} \equiv g[\ell] = \mathsf{H}$ and $g[\ell] = \mathsf{AA} \equiv g[\ell] = \mathsf{A}$. With the assumption of biallelic polymorphisms, haplotypes can generally be encoded as binary strings.

Throughout this text, we use following notation: n denotes the length of the condensed genotype, $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \{1, \ldots, n\}$ are consecutive blocks of loci, function $\mathsf{Pos} : \{1, \ldots, n\} \to \{1, \ldots, N\}$ maps loci of condensed representation to the true loci in the genotype.

Firstly, we define a representation of an inferred haplotype reconstruction. Let $g = h_1 \oplus h_2$ be the actual genotype and $\widehat{g} = \widehat{h}_1 \oplus \widehat{h}_2$ be the inferred haplotype configuration. Then let $\mathsf{Hap}_g(i)$ describe the inferred haplotype configuration g_{inf} with regard to the actual (known) genotype. We can define $\mathsf{Hap}_g(i)$ as follows

$$\mathsf{Hap}_g(i) = \begin{cases} \perp, & \text{if } h_1[i] = h_2[i]; \\ 1, & \text{if } \widehat{h}_1[i] = h_1[i]; \\ 2, & \text{if } \widehat{h}_1[i] = h_2[i]. \end{cases}$$

It is worth to notice that $\perp$ always marks the homozygotic loci. The haplotype inference has been flawless for a given sequence, if Hap_g contains only 1-s and no 2-s, or it contains only 2-s and no 1-s.

4.2 Error definitions and models

The simplest error rate definition arises from just making a distinction between correctly phased genotypes and incorrectly phased genotypes. We could simply define an error rate for inference on population G such that

$$\varepsilon_{naive}(G) = 1 - \frac{\#\{g : 1 \notin \mathsf{Hap}_g \vee 2 \notin \mathsf{Hap}_g\}}{|G|}$$

This error definition does not unfortunately capture the distinction between almost correct and completely wrong reconstructions. In many applications of inferred haplotypes a minor deviation from correct reconstruction would not affect the results considerably. We therefore seek a more detailed error definition, which provides a mechanism for measuring a correctness of a genotype resolution.

Single locus errors

A logical approach for defining a detailed error rate, would be counting correctly and incorrectly inferred SNPs in the genotype sequence. Technically we need to define the haplotype confinguration for a block of SNPs to understand whether individual constituent SNPs have been inferred correctly (with regard to the block)

(a) Correct haplotype phase

(b) Inferred phase with SNP error at locus 4

Figure 4.1: Example of single locus error

or not. In a specific case, the block could be the full sample sequence $\mathcal{S}$. We can define similar notation for a block of SNP-s $\mathcal{A}$ based on majority voting

$$\mathsf{Hap}_g(\mathcal{A}) = \begin{cases} 1, & \text{if } \sharp\left\{i \in \mathcal{A} : \mathsf{Hap}_g(i) = 1\right\} > \sharp\left\{i \in \mathcal{A} : \mathsf{Hap}_g(i) = 2\right\}, \\ 2, & \text{if } \sharp\left\{i \in \mathcal{A} : \mathsf{Hap}_g(i) = 1\right\} < \sharp\left\{i \in \mathcal{A} : \mathsf{Hap}_g(i) = 2\right\}, \\ \perp, & \text{if } \sharp\left\{i \in \mathcal{A} : \mathsf{Hap}_g(i) = 1\right\} = \sharp\left\{i \in \mathcal{A} : \mathsf{Hap}_g(i) = 2\right\}. \end{cases}$$

In words, a haplotype configuration for block $\mathcal{A}$ equals the configuration of its majority constituent SNP-s. It follows that by this definition any reconstruction is at least 50% correct.

In the following we omit the genotype index in the notation of Hap_g, while we will be considering one genotype at a time. We can thus define error vector with regard to the block $\mathcal{A}$, provided $i \in \mathcal{A}$

$$\mathsf{Loc}_{\mathcal{A}}(i) = \begin{cases} 0, & \text{if } \mathsf{Hap}(\mathcal{A}) = \mathsf{Hap}(i), \mathsf{Hap}(\mathcal{A}) \neq \perp, \\ 1, & \text{if } \mathsf{Hap}(\mathcal{A}) \neq \mathsf{Hap}(i), \mathsf{Hap}(\mathcal{A}) \neq \perp, \\ \perp, & \text{if } \mathsf{Hap}(i) = \perp \text{ or } \mathsf{Hap}(\mathcal{A}) = \perp. \end{cases}$$

i.e. $\mathsf{Loc}_{\mathcal{A}}(i)$ is set to 1 if there is explicit error and is set to $\perp$ or if we cannot determine whether there is an error or not. According to this definition we can define the error rate for a populatation of genotypes as

$$\varepsilon_{loc}(G) = \frac{1}{|G|} \sum_{g \in G} \frac{\#\{i : \mathsf{Loc}_{\mathcal{S}}(i) = 1\}}{\#\{i : \mathsf{Loc}_{\mathcal{S}}(i) \neq \perp\}}.$$

The described locus based approach resembles the widely used string edit distance calculation in bioinformatics, although without allowing insertions and deletions. Note that ε_{loc} is not compatible with ε_{naive} and generally $\varepsilon_{loc}(G) \ll \varepsilon_{naive}(G)$.

Turn errors

There is yet another way of looking at inference errors, noticing that the popular inference procedures largely rely on the assumption of *linkage disequilibrium* LD and models assuming crossing-over. For example, let us consider a block $\mathcal{A}$ of length 20 SNPs, with an SNP-wise error vector

$$\mathsf{Loc}_{\mathcal{A}} = 0\perp\perp000011\perp11100000\perp0.$$

(a) Correct haplotype phase

(b) Inferred phase with turn error somewhere between loci 4 and 7

Figure 4.2: Turn error illustration

Instead of saying that the inference process made 5 faults in the process, we might say that it only made 2 false turns. Locally, in a subblock of $\mathcal{A}_{(8,16)}$, the inference process has actually been correct.

When formalising **turn errors**, the simplest way would just be comparing an inferred SNP with the previous consecutive SNP:

$$\mathsf{Turn}(i+1) = \begin{cases} 0, & \text{if } \mathsf{Hap}(i) = \mathsf{Hap}(i-1), \\ 1, & \text{if } \mathsf{Hap}(i) \neq \mathsf{Hap}(i-1), \\ \perp, & \text{if } \mathsf{Hap}(i) = \perp \text{ or } \mathsf{Hap}(i-1) = \perp, \end{cases}$$

This unfortunately does not capture false turns between two heterozygote loci in cases, where there is a homozygote SNP between them. This simple definition would produce:

$$\mathsf{Hap}(\mathcal{A}) = 22\perp 11$$
$$\mathsf{Turn}(\mathcal{A}) = 0\perp\perp 0,$$

without picking up the false turn

$$\mathsf{Turn}(\mathcal{A}) = 0\perp 10.$$

To capture such errors, we need a way to compare with the previous heterozygotic locus on the sequence. For a locus i, let j be the first preceding heterozygous locus. Then

$$\mathsf{Turn}(i+1) = \begin{cases} \perp, & \text{if } \mathsf{Hap}(i) = \perp, \\ 0, & \text{if } \mathsf{Hap}(j) = \mathsf{Hap}(i), \\ 1, & \text{if } \mathsf{Hap}(j) \neq \mathsf{Hap}(i). \end{cases}$$

According to this definition the inference process has made a false turn only if the previous heterozygotic SNP has been inferred different.

We can use very similar definition for a set of consecutive blocks $\mathcal{A}_1 \dots \mathcal{A}_n$, for which $\mathsf{Hap}(\mathcal{A}_i)$ has been defined earlier

$$\mathsf{Turn}(\mathcal{A}_{i+1}) = \begin{cases} \perp, & \text{if } \mathsf{Hap}(\mathcal{A}_i) = \perp; \\ 0, & \text{if } \mathsf{Hap}(\mathcal{A}_j) = \mathsf{Hap}(\mathcal{A}_i); \\ 1, & \text{if } \mathsf{Hap}(\mathcal{A}_j) \neq \mathsf{Hap}(\mathcal{A}_i). \end{cases}$$

where $\mathcal{A}_j$ is the first preceding block to $\mathcal{A}_j$ such that $\mathsf{Hap}\mathcal{A}_j \neq= \perp$

One can think of the single locus error model as counting the number of mutations needed to obtain the true reconstruction from the inferred one. Following this logic, the turn error model expresses the difference by the number of crossover events.

$$\varepsilon_{turn}(G) = \frac{1}{|G|} \sum_{g \in G} \frac{\#\{i : \mathsf{Turn}(i) = 1\}}{\#\{i : \mathsf{Turn}(i) \neq \perp\}}.$$

Knowing the underlying biological models for the inference tools, it may be assumed that a mixed model would be applicable to study errors.

4.3 Observable error characteristic

In the previous section we defined two error models Loc and Turn, but neither of these errors can be determined by comparing the inference result with known haplotype reconstruction. Hence, we need to define an *error characteristic* that can be observed directly from data. An observable characteristic can be described by picking two heterozygotic SNPs i and j on a genotype and defining a **pair error** as

$$\mathsf{Pair}_g(i,j) = \begin{cases} 0, \text{ if } \mathsf{Hap}(i) = \mathsf{Hap}(j), \\ 1, \text{ if } \mathsf{Hap}(i) \neq \mathsf{Hap}(j). \end{cases}$$

where i and j are heterozygous loci, shortly denoted by $i, j \in \mathsf{Het}_g$.

If the inference process makes only single locus mistakes, then $\mathsf{Pair}(i,j) = 1$ means, that it failed to produce correct configuration either in i or j, but didn't fail in both. If this is true and the errors in individual loci are independent, there is an equal probability of observing $\mathsf{Pair}(i,j) = 1$ regardless of i and j or the distance between them. A calculation formula for an error rate can be given as

$$\varepsilon_{pair}(G) = \frac{1}{|G|} \sum_{g \in G} \frac{\#\{(i,j) \in \mathsf{Het}_g : \mathsf{Pair}(i,j) = 1\}}{(|\mathsf{Het}_g| - 1)(|\mathsf{Het}_g| - 2)/2}.$$

Let ε_{loc} be a uniform probability that inference process makes a fault at any given locus and let us assume that the error characteristic follows the pure single locus error model. Thus, the probability of observing a pair error would be

$$\varepsilon_{pair} = 2\varepsilon_{loc}(1 - \varepsilon_{loc}). \tag{4.1}$$

On the other hand, if the inference process would follow the turn error model, then $\mathsf{Pair}(i,j) = 1$ would require an odd number of false turns between i and j. If the total physical length of the sample sequence is short with regard to the recombination rate, we can discard the possibility of back-turn and assume that the probability of observing a pair error depends also on the distance between the two loci. Let us be reminded from Turn error discussion, that this error model expects haplotype inference method to capture crossing over events in the population.

The Turn errors are due to missed or mistakenly assumed crossing overs. As the probability of crossing over events are dependent on the recombination rate in population, then so are the Turn errors. Under the assumption that at least some of the errors are due to uncaptured crossing overs, the probability of which relies on the distance d, we would need to express the error rate as

$$\varepsilon_{pair}(G,d) = \frac{1}{|G|}\sum_{g\in G}\frac{\#\{(i,j)\in \mathsf{Het}_g : d(i,j)=d \text{ and } \mathsf{Pair}(i,j)=1\}}{\#\{(i,j) : d(i,j)=d\}}.$$

Several different distance measures can be used in place of the abstract $d(i,j)$. We will be using the following specific distances:

- d_{bp}: genomic distance, the number of base pairs on the physical DNA sequence between i and j, formally $d = \mathsf{Pos}(j) - \mathsf{Pos}(i)$.
- d_{het}: number of heterozygotic loci in the sample for the genotype, $d_{het}(i,j)$ equals the number of heterozygous loci between i and j.
- d_{loc}: number of loci between i and j in the sample as $d = j - i$

If we assume, that the errors arise from a pure turn error model, the process needs to make an odd number of mistaken turns along the sequence. Let $d = j - i$ be a number of potential turns on the sequence (e.g. a number of loci between i and j) and ε_{turn} describe a uniform turn error rate.

$$\begin{aligned}\varepsilon_{pair}(d) &= \binom{d}{1}\varepsilon_{turn}(1-\varepsilon_{turn})^d \\ &\quad + \binom{d}{3}\varepsilon_{turn}^3(1-\varepsilon_{turn})^{d-3} + \dots\end{aligned} \tag{4.2}$$

$$\varepsilon_{pair}(d) = \sum_{k=0}^{\lfloor d/2\rfloor}\binom{d}{2k+1}\varepsilon_{turn}^{2k+1}(1-\varepsilon_{turn})^{d-(2k+1)} \tag{4.3}$$

Using Newton's binomial formula and noticing that $(x+y)^n-(-x+y)^n$ eliminates odd-powered x-terms from the binomial series, we can simplify

$$\begin{aligned}\varepsilon_{pair}(d) &= 1 - \frac{(\varepsilon_{turn}+(1-\varepsilon_{turn}))^d + (\varepsilon_{turn}-(1-\varepsilon_{turn}))^d}{2} = \\ &= \frac{1-(1-2\,\varepsilon_{turn})^d}{2}.\end{aligned} \tag{4.4}$$

As the pair error characteristic is simple and robust, we will use this characteristic in further empirical study of inference accuracy. We will also try to fit the abovementioned distance models and learn more about the distance-accuracy dependency.

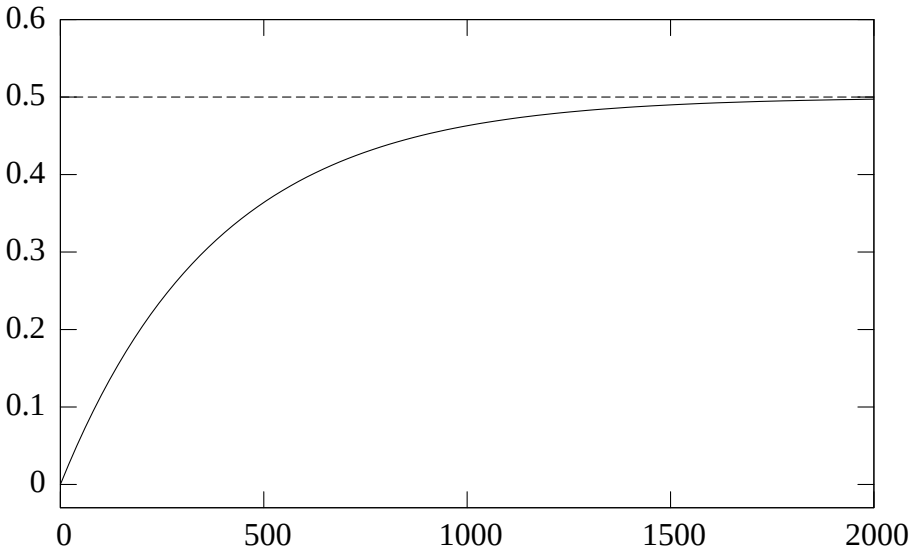


Figure 4.3: Pair error probability $\varepsilon_{pair}(d_{het})$ in case of a pure turn error model. In this illustration $\varepsilon_{turn} = 0.0011$, empirically estimated from HapMap population. The curve conforms to the intuition that as the distance between the considered loci grows, the probability of inferring the correct phase approaches $1/2$. We see that at this characteristic turn error rate, there is not much reason for inferring haplotypes in segments larger than 800 heterozygotes. The same may not hold for different populations or SNP frequencies.

4.4 Datasets

There are only very few biological datasets available that contain both genotypes and biologically known haplotypes. The population sizes and sequence lengths for such biological experiments are very limited. We therefore also consider some alternative approaches to prepare datasets for testing inference accuracy.

Haplotypes from pre-phased genotype data

A method we have used for creating a semi-synthetic dataset involves haplotype inference in data preparation:

1. computationally infer haplotypes from genotypes for population G; and
2. simulate one generation of random mating within population G without recombinations or mutations.

The approach relies on the belief that inferred haplotypes approximately represent the underlying haplotype distribution

$$\Pr[H] \sim \Pr[H|G].$$

We acknowledge, that it is usually not a good idea to prepare data with the same method that is later used for testing the accuracy of the method. We have not conducted specific experiments to assess such bias in our pre-phased datasets.

We have used two HapMap.org [Con05] populations for input data—Central European (CEU, 89 genotypes) [Hap06a] and Japanese (JPT, 43 genotypes) [Hap06b]. We have picked 40 disjunct 300-SNP blocks uniformly from 18, 19, 20, 21 chromosomes, 10 blocks from each chromosome. Average genomic distance of consecutive SNPs in our HapMap.org datasets is 2.5 kilobases.

Synthetic dataset—simulated population of haplotypes

A widely used approach is to use synthetic datasets simulated by one of the mathematical models, which we described in the previous chapter. There are various tools available for such simulations and we chose Hudson's simulator `ms` described in [Hud02]. Out of the various models, that the tool supports, we picked the basic neutral model with constant recombination rate, constant population size and infinite mutation sites.

Our aim in creating the synthetic dataset was to simulate a haploid population similar to the HapMap populations. We wanted to verify `phase` error rates with another dataset not prepared using `phase` itself, in order to verify that there is no considerable bias in pre-phased HapMap datasets.

To create a synthetic dataset similar to the existing HapMap datasets, we estimated the population parameters from the HapMap CEU population genotypes:

$$\widehat{\theta} = 6.0$$
$$\widehat{\rho} = 0.00018$$

using Felsestine's `recombine` program [KYF00]. This tool simulates random haploid ancestry trees on top of the population sample. It uses Metropolis-Hastings algorithm to infer the most likely trees and compute estimated population parameters.

Our synthetic dataset consists of 10 haploid simulated samples [Syn06] and respective genotypes obtained via random mating.

Perlegen biological dataset

Patil *et al* at Perlegen [PBH$^+$01] have sequenced 24047 haploid loci from the 21st chromosome. The data files provided by Patil present the sequences as haplotype blocks for 20 individual haplotypes, but using some scripting, these can be combined into 20 long haplotype sequences.

The two problems arising with the biological dataset are a great percentage of undetermined loci, about $\sim 21.7\%$, and the modest number of haplotypes available. As we would like to use 300 loci segments extending over 750 kilobases similar to HapMap average SNP distance, it would be normally quite unlikely to observe more than one copy of a distinct haplotype over such long region in a relatively small population. We therefore should not use any haplotype twice when we generate haplotypes by random mating.

By carefully selecting the 300 SNP regions to be cut out from the long sequence, we could bring down the amount of missing data to $\sim 16\%$. We also

expanded the set of haplotypes artificially, by randomly duplicating 10 haplotypes in the sample. As a result, we randomly mated the haplotypes and and produced 15 genotypes. We cut out five 300 SNP regions from the genotypes, thus creating 5 input samples [Per06].

4.5 Empiric error measurement

Out of the several proposed models, datasets and available tools we focused on testing the phase software on HapMap datasets. Our preference of phase was mainly due to the lack of reliable alternative tools, which could cope with the long genotype sequences that we were particularly interested in. Expectation-maximisation based algorithms, for example, are not practical for sequences longer than 20 SNPs. HapMap consortium has also chosen phase to analyse their data [Con05]. From experiments we inferred pair error charcteristic $\widehat{\varepsilon}_{pair}$, as it gives a practical error measure without making assumptions about the inference model. As discussed above, an observed pair error rate distribution also allows to make some conclusions about the underlying error model.

Following our earlier definition, we would like to estimate the mean error rate conditional on the distance

$$\mathbf{E}_{\varepsilon}(\Pr\left[\mathsf{Pair}(i,j) = 1 \mid d(i,j) = d\right]) = \varepsilon_{pair}(G,d),$$

where distance measure can be one of d_{bp}, d_{het} and d_{loc} as described above.

It should be acknowledged that the results also depend on how the datasets have been combined—how densely the loci in the sample are positioned on the genotype, how heterogeneous is the population, etc.

The experiment is set up as follows. First, we infer haplotypes for input datasets and find all pairs of heterozygotic SNPs on genotypes in every dataset. We then determine d_{bp}, d_{het} and d_{loc} for each pair and compare inference result to known true haplotype configuration, determining $\mathsf{Pair}(i,j)$ for every pair. Finally we éstimate the error rate by computing average error probability for every distance.

As the inference of one 300 loci HapMap sample takes on average 15 hours to complete on a modern PC with standard phase configuration, the total CPU time needed for all 80 HapMap samples was about 1200 hours or 50 CPU days. Similar extent of processing resource was needed to set up the known-haplotype datasets in the first place. We took advantage of parallel processing in *Estonian Grid* computer clusters in order to engage sufficient computational power for inferring haplotypes in the samples. The error measurement experiment finally took about 48 hours on the grid.

4.6 Empirically observed error rates

The results obtained from the experiment are shown on Figure 4.4. It can be visually seen from the three curves, that there is less variation in the probabilities plotted by the distance measure d_{het}. Intuitively distance in heterozygote loci

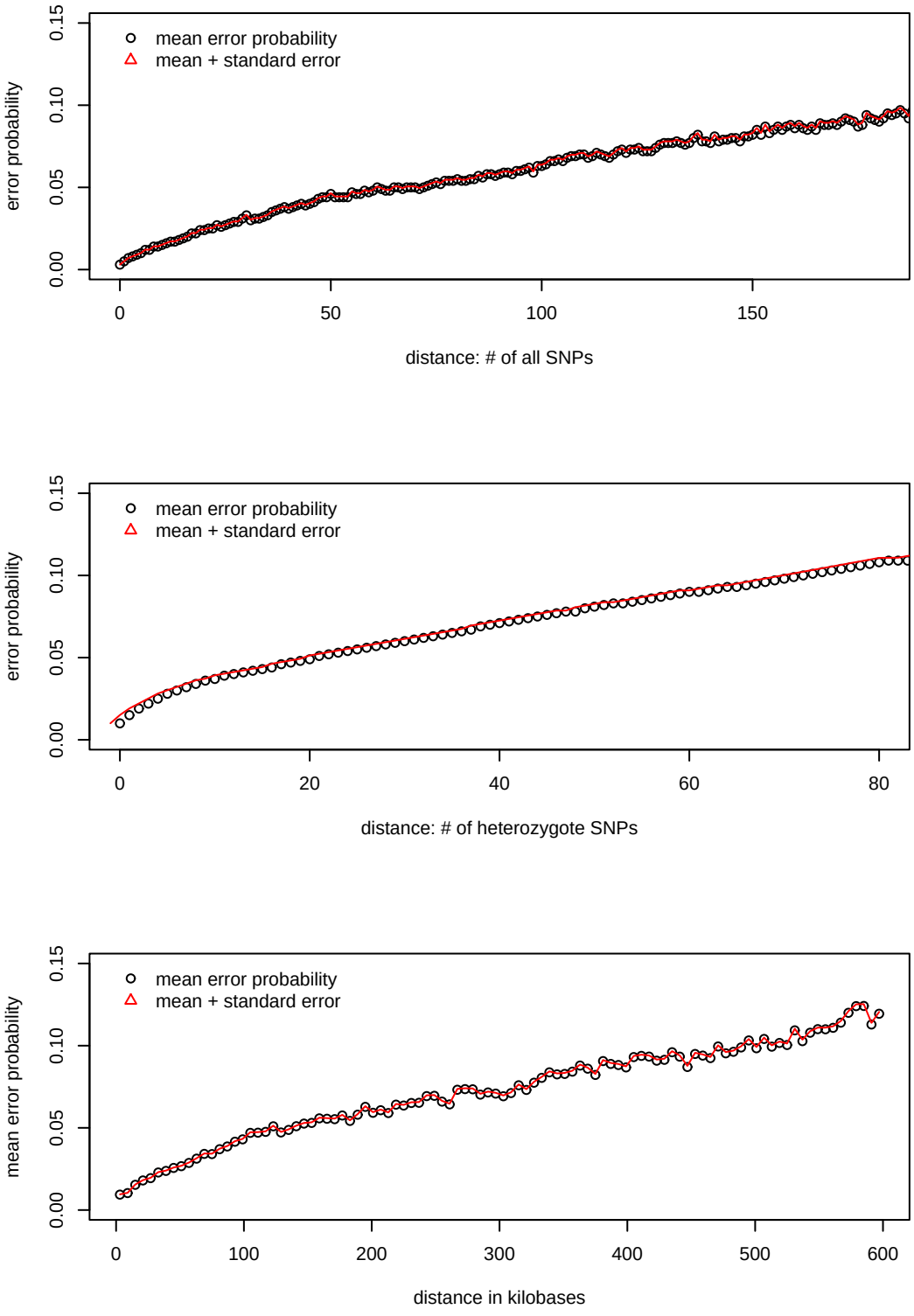


Figure 4.4: Measured `phase` inference error rates in pre-phased HapMap.org datasets. The curves describe $\mathsf{Pair}(i, j)$ error dependent on distance measures d_{loc}, d_{het}, d_{bp} from top down respectively. As the samples where large containing substantial number of individual pairs, the observed standard error is negligible.

appears to be a more natural parameter to the underlying error model at least when considering `phase`. We also notice that for loci, which are 5 to 75 heterozygotes apart, the estimated error probability is almost linear to d_{het}. With simple curve fitting in, we obtained

$$\widehat{\varepsilon}_{pair}(G \mid d_{het} \in [5, 75]) = 0.001 * d_{het} + 0.024. \tag{4.5}$$

This approximated formula may not apply for other populations, where genomic diversity and locus density may be considerably different. To derive a more robust error rate and establish quantified associations between error rates and known population parameters, it would require conducting further studies with different populations and locus densities. Linkage disequilibrium is generally stronger in the protein encoding parts of the genome and weaker in non-encoding regions. We suspect that such variance in LD may also add to the variation of error rates. Due to time constraints, these studies fell out of scope for this work.

The results clearly exhibit that the observed error rates are dependent on distance, which rules out the prevalence of a single locus error model for `phase`. The turn error model could be well fitted to the results with the reservation, that once there has occured a false turn, it is highly unlikely (at least with our sequence lengths) to observe another false turn on the same genotype, which would correct the previous turn.

It could be argued, that the inference process is designed to capture recombination events in the population and errors tend to occur, when it does not pick up rare events, which represented in only one or two haplotypes. It would be easy for the process to capture common events, because there is more support for these in the data. These missed out recombinations (false turns) may just be so recent in the population and they haven't propagated to other genotypes. As they are recent, no other recombinations have yet occured in close enough proximity that would have corrected the false turn.

4.7 Error rate variance under specific conditions

Although the general picture was fairly consistent, we paid extra attention to cases, where we believed that inferring the correct phase would be more difficult. As our samples contained a large number of heterozygote pairs, these special cases would easily average out and remain unnoticed in the general experiment.

Errors on the edges of a sequence

As the aim of this work is to propose an enhanced partition-ligation scheme with overlapping segments, we were particularly concerned with a potentially increased error rate on either end of the inferred segment. Knowing that the inference models rely on the concept of linkage disequilibrium, then intuitively there is less information in the sample for loci that are situated close to the edge of the sequence, since in the middle part of seguence algorithms can use correlation (LD) between previous and forthcoming loci.

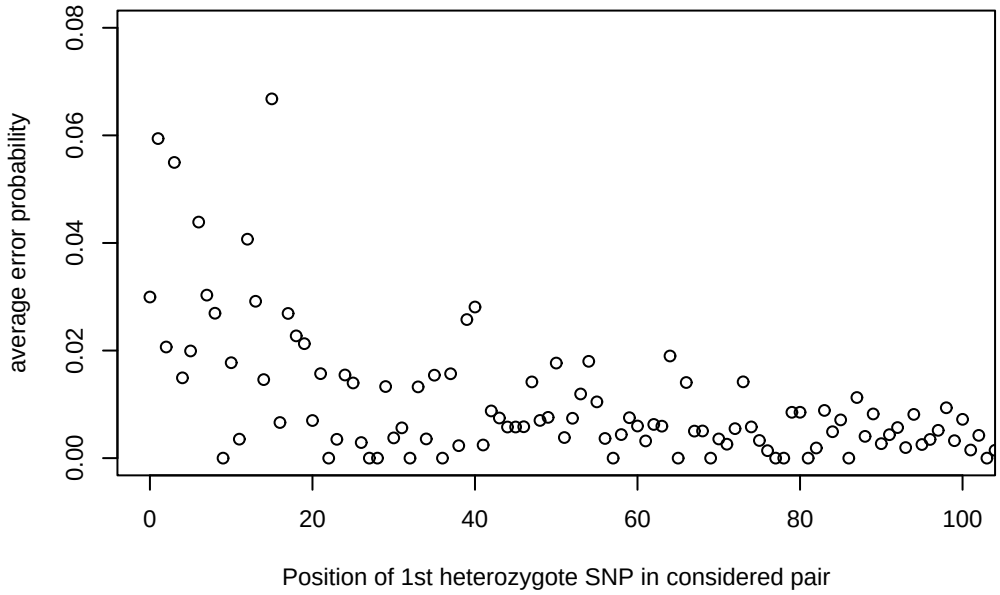


Figure 4.5: Error probabilities are higher on the edges of the input sequence. These results may also affect the accuracy of our overlapped ligation scheme.

We therefore conducted another experiment on [Hap06a] dataset to study the error probabilities with regard to the position of loci in the input sequence. Specifically, we were interested in the pair error rate among such pairs of heterozygote loci, where the first locus of the pair is situated near the start of the sample sequence.

$$\Pr\left[\mathsf{Pair}(i,j) \mid i\right]$$

The results suggest, that there generally is remarkably higher probability to observe errors near the start of the input sequence. Figure 4.5 plots average pair error rate by the position of the first heterozygous locus of the pair, hinting that if one of the loci is on the very edge of the sequence, then the estimated error probability is about three times the average.

In another experiment, illustrated in Figure 4.6, we observed error probability by the position of the turn error. For this purpose, we considered such heterozygote pairs that were 3 to 5 heterozygotes apart and at the same time 10 to 30 loci apart in the sample.

$$\Pr\left[\mathsf{Pair}(i,j) \mid d_{het} \in [3,5] \text{ and } d_{loc} \in [10,30]\right]$$

This is because we needed a homogeneous sample in terms of distance-dependent error. We added some flexibility to maintain reasonably large sample size. Among these considered pairs, we assumed that the potential error is made at the midpoint $\mathtt{pos}(\mathsf{Pair}(i,j)) = (i+j)/2$. We see again that the turn errors are considerably more likely to occur at the start of the sample and also more probable at the end of the sequence. We can also see some "jumps" in error rates in the middle

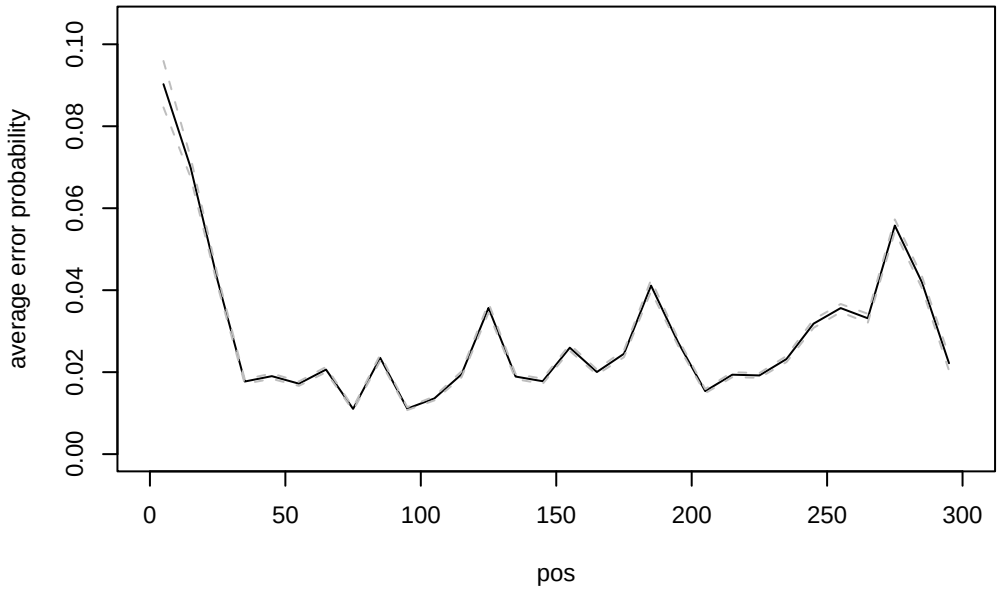


Figure 4.6: Error probabilities are plotted by the estimated position of turn-error occurance. Turn errors are considerably more likely to occur at the start of the sample and also more probable at the end of the sequence. We can also see some "jumps" in error rates in the middle part of the sample. It may be argued that these are due to the internal hierarchical partition-ligation method used in `phase`. Plotted dots represent average error probability, while the plotted line is a sum of mean and standard error. Our datasets are large enough that standard error almost vanishes.

part of the sample. It may be argued that these are due to the internal hierarchical partition-ligation method used in `phase`.

Effects of long homozygous regions

We were also curious, what happens if there is a long block of homozygotes in the sequence – can the inference process still pick up the correct phase for a pair of heterozygotic loci separated by a homozygote region? Following our notation we compare

$$\widehat{\varepsilon}_{pair}(G, d_{loc}) \text{ with}$$

$$\Pr\left[\mathsf{Pair}(i,j) \mid d_{loc}, \text{ no heterozygotes between } i \text{ and } j\right].$$

Results shown in Figure 4.7 suggest that error rate among such pairs, which are separated by a long region of homozygotes, is two times higher for longer homozygote regions. As the number of such pairs is not large for longer regions, the variance in error rates is also considerably higher.

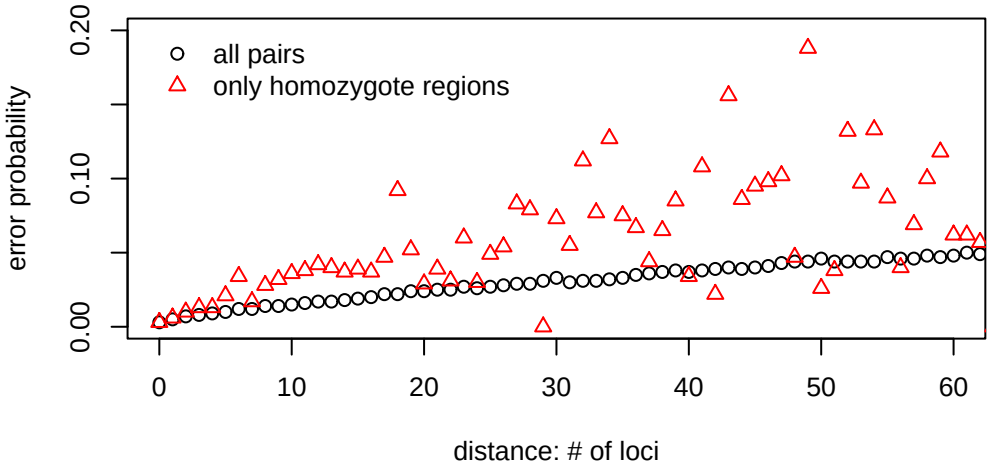


Figure 4.7: Error probability among such pairs, which are separated by a long region of homozygotes, plotted by the length of homozygote region. Error rates in this experiment were about twice above average. As the number of such pairs is not large for longer regions, the variance in error rates is also considerably higher.

Results with the synthetic dataset

For assurance that `phase` error rates behave similarly when obtained from other datasets, we ran a small test with our synthetic simulated dataset.

We can see from Figure 4.8 that the distance dependent distribution of error rates is similar to the HapMap dataset, although the mean error rates are 2-3 times higher. We noticed that as we were using two different softwares for estimating population parameters (`recombine`) and for simulating according the estimated paramters (`ms`), the interpretation and use of the paramaters, particularly recombination rates, was slightly different by Hudson and Felsestein. For example, the share of heterozygotic loci was remarkably greater in the simulated dataset than the original HapMap genotypes. This is a probable reason why `phase` performs significantly worse on our synthetic data. We believe that it would be possible to eliminate these differences by running a few estimation-simulation-reestimation cycles.

4.8 Empirical error model

We have already shown the simple approximate formula (4.5) for analytically estimating characteristic pair error rate on a limited sequence length. Let us now

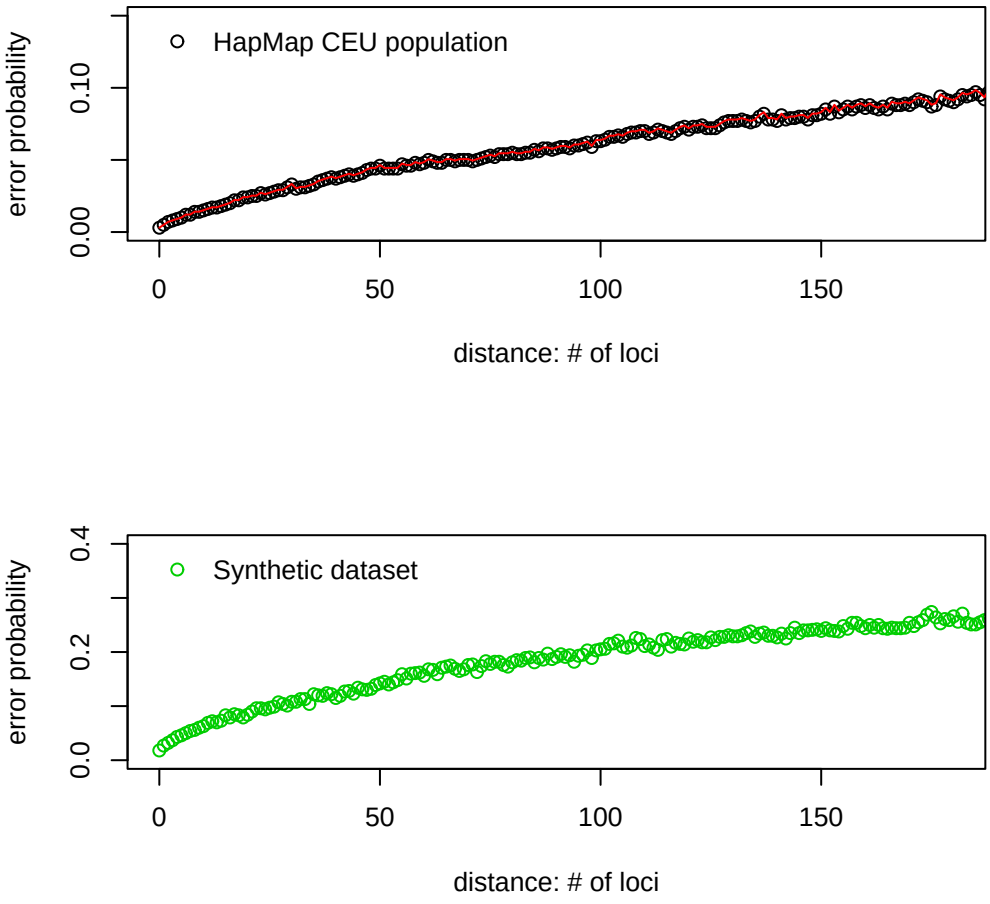


Figure 4.8: Pair error rates by distance measure d_{loc} for HapMap.org population (top plot) and synthetic coalescent simulted dataset (below plot). The curves are very similar in terms of shape, but differing on error scale. We believe that this is due to different population and sampling parameters in the synthetic dataset.

propose a mixed model that can be fitted to produce the observed rates, such that

$$\begin{aligned}\varepsilon_{pair} &= \lambda\, f(\varepsilon_{turn}) + (1-\lambda)\, g(\varepsilon_{loc});\\ \varepsilon_{pair} &= \lambda\frac{1-(1-2\,\varepsilon_{turn})^{d_{het}}}{2} + (1-\lambda)\, 2\varepsilon_{loc}(1-\varepsilon_{loc}),\end{aligned} \tag{4.6}$$

where (4.6) has been obtained using equations (4.1) and (4.4) given for pure models. We used nonlinear regression method `nls()` from `R` statistics tool, to find an approximated solution, best matching our observed error rates. The nonlinear regression in `R` used Gauss-Newton algorithm for this least squares parameter estimation problem.

$$\begin{aligned}\lambda &\approx 0.95;\\ \varepsilon_{loc} &\approx 0.47;\\ \varepsilon_{turn} &\approx 0.00124.\end{aligned}$$

The residual sum-of-squares for this approximation in the region $d_{het} \in [0; 119]$ is 0.00098. We can formulate the same problem without the λ parameter as in equation (4.7), assuming that `phase` makes both type of mistakes independently.

$$\varepsilon_{pair} = \frac{1-(1-2\,\varepsilon_{turn})^{d_{het}}}{2} + 2\varepsilon_{loc}(1-\varepsilon_{loc}), \tag{4.7}$$

An approximate solution to this model returns

$$\begin{aligned}\varepsilon_{loc} &\approx 0.0121;\\ \varepsilon_{turn} &\approx 0.00117.\end{aligned}$$

for parameter values with similar sum-of-squares approximation error.

We use the latter model (4.7) to write the pair error characteristic calculation formula

$$\varepsilon_{pair} = \frac{1-(0.9976)^{d_{het}}}{2} + 0.024, \tag{4.8}$$

while keeping in mind, that we are only considering distances $d_{het} \in [5, 75]$, specific populations and locus densities. We have illustrated the fitting of (4.6) on Figure 4.9. The fitting of (4.7) has no notable difference and is therefore omitted.

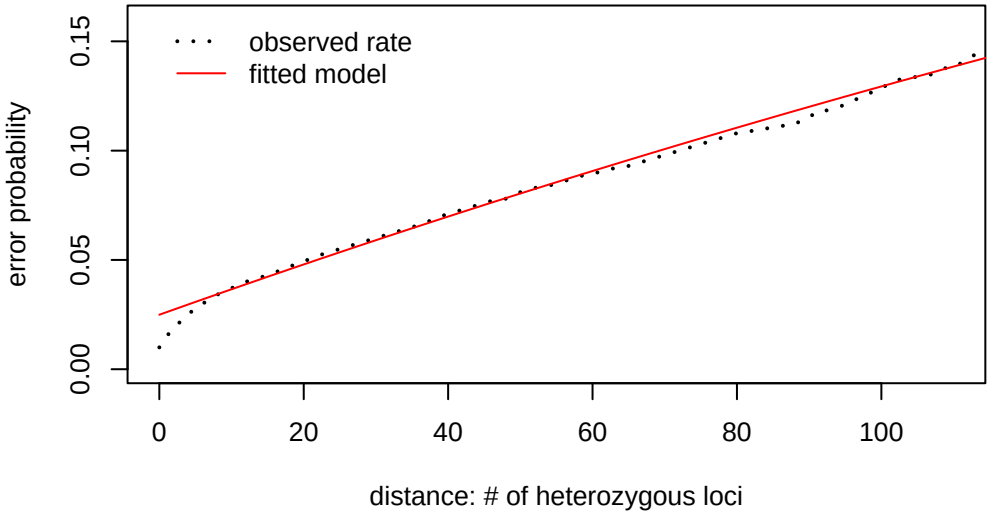


Figure 4.9: Our proposed mixed model $\varepsilon_{pair}(d_{het})$ from equation (4.6) plotted against observed rates in HapMap.org population.

Chapter 5

Proposed Enhancement to Partition-Ligation

Having now described the known inference methods and set up a framework for measuring and estimating inference error, we are about to propose an enhanced partition-ligation (PL) technique. This enhanced PL involves creating an overlapped partitioning on a given sequence. We then use the heterozygotic loci in the overlaps to choose a correct phase at the ligation step. The method relies on the assumption that a pair of independently phased neighbouring segments can be ligated using the known correct phase of heterozygotic loci in the overlap.

In this chapter we describe an algorithm for obtaining an optimal partitioning for a given genotype population. We characterise the distribution of segment lengths with regard to the minimum number of heterozygotic loci in overlaps. The ligation step is set up in two phases. In the first ligation phase we try to use heterozygotic loci in overlaps to determine the correct phase. Having the error model at hand, we can also calculate the confidence for all these ligations. Unfortunately not all segments can be ligated using the overlaps. We therefore use the calculated confidence rates in the second ligation phase, where we reprocess the undetermined ligation sites using refined statistical methods. The overlap-determined ligations along with their confidence ratios are taken as known data in the refined ligation phase. We aim to illustrate the various steps with practical experiments on the HapMap.org populations.

5.1 Partitioning

There are several ways to approach the partitioning tasks. It is intuitive that the goodness of a partitioning is higher when more individual overlaps contain heterozygote loci. If we consider computational cost, then inferring haplotypes in long segments is costly. In principle, there can be three-fold and higher degree overlaps, but this again adds to the computational expense. We aim to keep the model simple and describe the method for a fully overlapped two-fold case as illustrated in Figure 5.1. We seek such a partitioning $\mathcal{A}_1 \ldots \mathcal{A}_m$ on a sequence $\mathcal{S}$ for population G, which minimises the lengths of individual segments; and for every pair of overlapping segments, maximises the number of such individual geno-

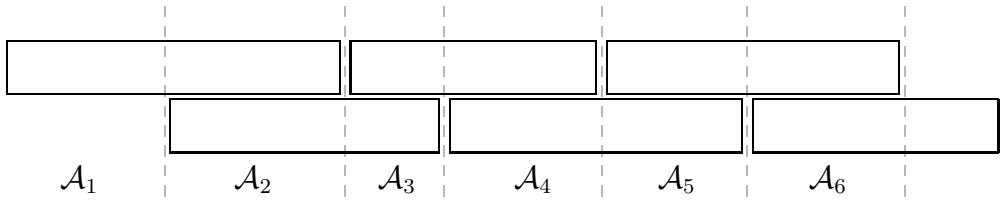


Figure 5.1: We have chosen a fully overlapped two-fold partitioning model for its simplicity. The sequence $\mathcal{S}$ is first partitioned into optimal overlaps $\mathcal{A}_1, \ldots, \mathcal{A}_m$. Segments for inference input are then formed as pairs of consecutive overlaps.

types in population G that contain heterozygotic loci in the overlapping region. The segment length in our model is determined by the length of overlaps with the consecutive segments. Thus, we can focus on assessing the goodness of overlaps. Instead of attempting to partition the sequence into overlapping segments, we can seek a partitioning into consecutive overlaps.

In the following we describe a model for estimating the number of genotypes in a population that are heterozygous in some segment $|\mathcal{A}| = d$.

5.1.1 Heterozygosity model

For a polymorphic locus that has most even allele frequency, e.g 50% chance observing either A or G, it can be estimated, that about 50% of individuals of an arbitrary population would be heterozygotic in that locus. If the allele frequencies were 90% to 10%, the chances of observing a heterozygote would drop to 18%.

Let p note the average probability of observing a minor allele. The probability of observing a heterozygous SNP is then just

$$\Pr[\text{ heterozygous locus }] = 2p(1-p).$$

Consider a sequence of length n. The percentage of individuals having h heterozygous loci on the sequence, follows the binomial distribution

$$\Pr[\, h \text{ heterozygotes in } n \text{ loci }] = \binom{n}{h}(2p-2p^2)^h(1-2p+2p^2)^{n-h} \quad ,$$

$$\Pr[\text{ at least } h \text{ heterozygotes in } n \text{ loci }] = 1 - \sum_{k=0}^{h-1}\binom{n}{k}(2p-2p^2)^k(1-2p+2p^2)^{n-k} \quad .$$

Using this formula and assuming uniform distribution of heterozygotes, we can estimate that for uniform 5% minor allele frequency we need a region of 31 SNPs to ensure at least 2 heterozygous loci in 80% of genotypes. The uniform distribution of heterozygotes, however, is not adequate for most real genotypes and this simple mathematical model only can provide very basic estimations.

5.1.2 Partitioning algorithm

To find the optimal partitioning on some sequence for a given population of genotypes, we constructed the following algorithm. Borrowing from dynamic pro-

gramming technique we noticed that we simplified the task by fixing the following input parameters:

1. The least number of heterozygous loci h present in every overlap;
2. A threshold t—percentage of individual genotypes, required to have h heterozygotic loci in a given overlap.

Let $\mathcal{S}$ denote the given sequence of loci $\mathcal{S} = \ell_1, \ldots, \ell_n$, for which we have SNP data for some population $G = g_1, \ldots, g_m$. Algorithm 5 starts constructing the partitioning from the start of the sequence, iteratively adding loci to the current segment until the number of genotypes with required heterozygosity exceeds the input threshold. As soon as the threshold is met, the segment is finished and the algorithm starts creating a new segment from the next locus on the sequence.

5.1.3 Experimental results

We used raw genotypes from HapMap.org to assess how well is it possible to partition sequences. For this purpose we extracted 20 datasets from chromosomes 18, 19 and 20, all of which contained 3000 polymorphic loci.

After running our partitioning algorithm on the sequences we plotted the cumulative distribution function (CDF) curves on Figure 5.2 to describe the distribution of segment lengths. The lengths are described with regard to combinations of thresholds t, population sizes $|G|$ and required number of heterozygotes h per overlap. The lengths are for final segments, really a sum of two neighbouring segments as produced by the algorithm $\#\mathsf{loci} = |\mathcal{A}_k| + |\mathcal{A}_{k+1}|$.

We should notice from Figure 5.2, that there is little value in requiring 90% threshold, because there is a fairly high chance (10% probability), that the partitioning would contain a block larger than 300, which is already considerably expensive to phase. In our further experiments, we have chosen $h = 1$ and set the threshold to 80%. At these values, the partitioning should produce less than 5% of segments that are longer than 120 loci. Recall from Figure 4.7, that error rates accross long homozygous regions are 2–3 times higher than average. Therefore, it is not practical to extend segment lengths to more than 100 loci, because the error probability between the ends of such segments will be high. The distribution of segment lengths for such partitioning on [Hap06a] data is described in Figure 5.8.

5.2 Ligating segments by overlapping

We described our enhancement to ligation procedure earlier in section 3.5. In the following sections we provide an algorithm for ligating overlapping segments, together with confidence and accuracy measurements.

5.2.1 Error characteristic

We use pair error characteristic to $\mathsf{Pair}(i, j)$ to measure the accuracy of overlapped ligations. Let $\mathcal{X} = \mathcal{A}_{j-1}\mathcal{A}_j$ and $\mathcal{Y} = \mathcal{A}_j\mathcal{A}_{j+1}$ be the inferred segments, such that

Algorithm 5: Optimal partitioning algorithm.

Input: Genotype sequences G, a heterozygocity parameter h and threshold t.

Output: An optimal partitioning $\mathcal{A}_1, \ldots, \mathcal{A}_j$ with regard to the threshold parameter.

function PARTITION(G, h, t)
Initialise $\mathcal{A}_1 = \emptyset$
$j = 1$
for $i = 0$ **to** $|\mathcal{S}|$
  Add next locus to $\mathcal{A}_j = \mathcal{A}_j \cup \ell_i$
  Let $G' = \{g \in G : \text{HZGCOUNT}(g, \mathcal{A}_j) \geq h\}$
  if $|G'|/|G| \geq t$ **then**
    Finished with this segment, starting next one $j = j + 1$
  end if
end for
return $\mathcal{A}_1, \ldots, \mathcal{A}_j$
end function

function HZGCOUNT($\mathcal{A}, g$)
$h = 0$
for $k = 1$ **to** $|\mathcal{A}|$
  if k-th locus is heterozygous in g
    $h = h + 1$
  end if
end for
return h
end function

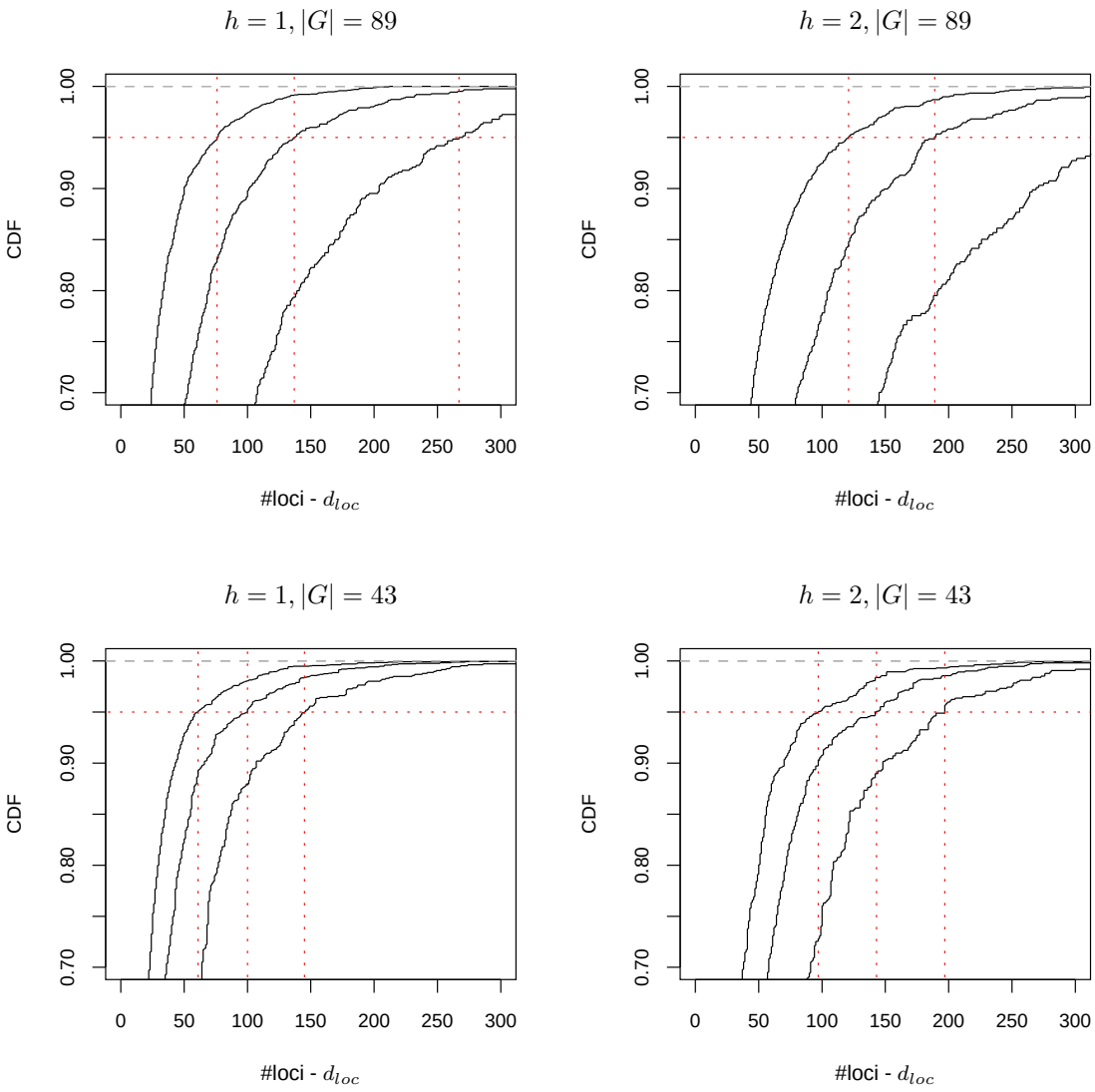


Figure 5.2: Distribution of segment lengths $t_1 = 0.7$ (highest curve) , $t_2 = 0.8, t_3 = 0.9$ (lowest curve). The segment lengths refer to the lengths of two consecutive, i.e. the full lengths for inference input. For example, the top-right plot reads that when requiring 2 heterozygores per overlap and all overlaps to be heterozygous in at least 80% genotypes, then an estimated 5% of the segments exceed 180 loci.

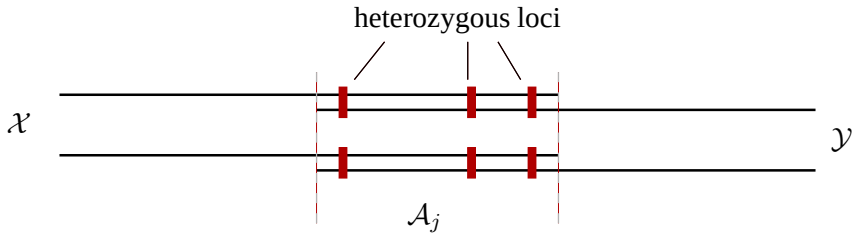


Figure 5.3: Ligation of two overlapping segments. For this example, the correct phase is the one suggested by at least two heterozygous loci in the overlap.

they overlap in $\mathcal{A}_j$. Let a be the first heterozygous locus in $\mathcal{A}_{j-1}$ and z be the last heterozygous locus in $\mathcal{A}_{j+1}$ for some genotype g.

We define

$$\mathsf{Lig}_g(\mathcal{X},\mathcal{Y}) = \mathsf{Pair}_{g[\mathcal{X}\mathcal{Y}]}(a,z).$$

as a ligation error characteristic. Note that Lig can be caused either by a false ligation, a turn or a locus error inside one of the segments. We can also define distance measures for segments

$$d(\mathcal{X},\mathcal{Y}) = d(a,z).$$

Using the flavors of the measure d_{loc}, d_{het}, d_{bp} we can express the error probability

$$\varepsilon_{lig}(G,d) = \frac{1}{|G|}\sum_{g\in G}\frac{\#\{\mathcal{X},\mathcal{Y} : d(\mathcal{X},\mathcal{Y}) = d \text{ and } \mathsf{Lig}_g(\mathcal{X},\mathcal{Y}) = 1\}}{\#\{\mathcal{X},\mathcal{Y} : d(\mathcal{X},\mathcal{Y}) = d\}}.$$

Note that due to our partitioning model, there may not be heterozygous loci in everly overlap and therefore the characteristic Lig_g only applies to those neighbouring segments that have heterozygotes not belonging to their common overlap. We do not assume that there is any bias related to such setup.

5.2.2 Confidence

We hereby propose a confidence measure for deciding the phase of a particular ligation. As we are using `phase` as our core inference engine, this tool outputs confidence levels for each locus along with the inferred haplotype reconstruction.

Calculation using prior confi dence

Let $0 \leq C_g(i,\mathcal{A}) \leq 1$ denote the confidence of inferring certain phase at locus i on genotype g, while processing segment $\mathcal{A}$. Let us use these prior confidence levels to define $\mathsf{Conf}_g(\mathcal{X},\mathcal{Y})[i]$ for overlapping segments $\mathcal{X} = \mathcal{A}_{j-1}\mathcal{A}_j$ and $\mathcal{Y} = \mathcal{A}_j\mathcal{A}_{j+1}$, where i is a heterozygous locus in the overlap $i \in \mathsf{Het}(\mathcal{A}_j)$.

$$\mathsf{Conf}_g(\mathcal{X},\mathcal{Y})[i] = C_g(i,\mathcal{X})C_g(i,\mathcal{Y}) + (1 - C_g(i,\mathcal{X}))(1 - C_g(i,\mathcal{Y})) \qquad (5.1)$$

Let $\mathsf{Het}(\mathcal{A}_j)$ contain heterozygous loci in the overlap and let $S \subseteq \mathsf{Het}(\mathcal{A}_j)$ be a subset of loci that support our ligation. Then we express the confidence of the

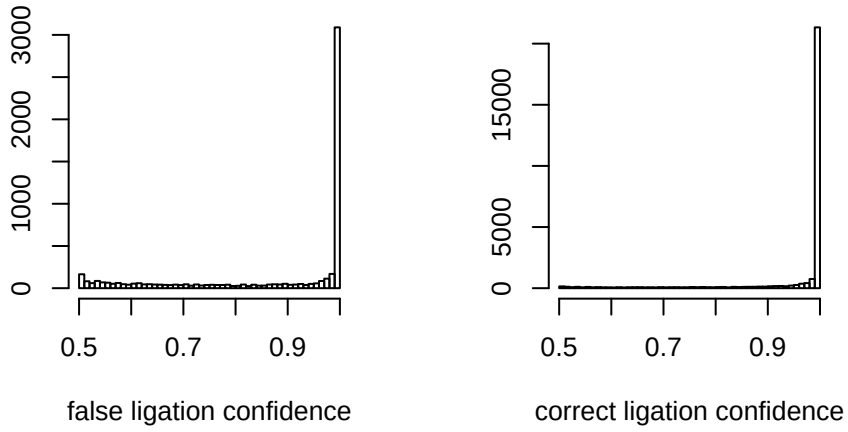


Figure 5.4: Histograms compare the confidence distributions of Conf for segments that were correctly ligated versus overlaps that were incorrectly ligated. Although there is high confidence for ligations in most cases, there appear more false ligations for confidences between $0.5 \ldots 0.9$. The results were obtained from running our ligation process on HapMap.org CEU population data. Heterozygosity threshold 0.8 was used in partitioning and one heterozygote locus was required in overlaps.

ligation as

$$\mathsf{Conf}_g(\mathcal{X}, \mathcal{Y}) = \frac{\prod\limits_{i \in S} c_i \prod\limits_{i \notin S} (1 - c_i)}{\prod\limits_{i \in S} c_i \prod\limits_{i \notin S} (1 - c_i) + \prod\limits_{i \notin S} c_i \prod\limits_{i \in S} (1 - c_i)}, \tag{5.2}$$

where we have used $c_i = \mathsf{Conf}_g(\mathcal{X}, \mathcal{Y})[i]$ for shorthand. The latter corresponds to the true ligation error probability if all c_i values are correct and errors occur independently. The fact that phase often suggests 1.0 confidence for some heterozygous loci, poses some trouble for the ligation confidence calculation. Even if a large majority of loci suggest a particular phase, it may therefore occur, that the confidence for both ligations may not be determined because of division by zero. To avoid this situation, we scale the prior phase confidence slightly down by replacing 1.0 with $1.0 - \epsilon$.

We use this same confidence measure in our ligation process. First, we choose a random phase of segments $\mathcal{X}$ and $\mathcal{Y}$. If the calculated confidence is above 0.5, we believe that we have chosen the right phase. If the confidence falls below 0.5, we flip $\mathcal{Y}$ around and this automatically inverts the supporting vs. non-supporting loci. The maximum value of $\max(\mathsf{Conf}, (1 - \mathsf{Conf}))$ represents our confidence in the ligation, with respect to phase confidence of individual locus-phases on segments.

Figure 5.4 compares the confidence distributions of Conf for segments that were correctly ligated versus overlaps that were incorrectly ligated. Although

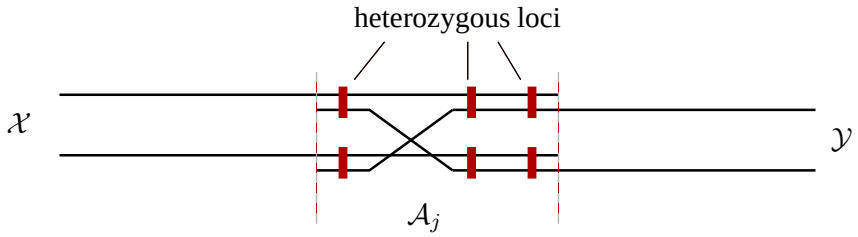


Figure 5.5: Ligation of two overlapping segments. In this example, the first heterozygous locus suggests a different ligation than the two remaining loci. There is reason to believe that there has been a turn error on one of the segments.

there is high confidence for ligations in most cases, there appear more false ligations for confidences between $0.5 \ldots 0.9$. It shows that for low confidence ligations, there is more probability for error than for high confidence ligations. This is well in accord with the intuitive notion of confidence.

Calculation using empirical error model

An alternative approach to ligation would be using our empirical error estimation models described earlier. This is a more general approach that can be used with other core inference tools, which do not output individual locus phase confidence. The basic idea is to use error estimates to calculate a probability for observing the correct phase of segments in overlap as opposed to the incorrect phase. We have earlier proposed two error models: turn-error model with characteristic $\varepsilon_{turn}(d)$ and a single locus error model described by ε_{loc}.

According to the turn model, we can estimate the probability of a mistaken turn on either segment. Unfortunately, this probability is fully symmetric according to our model and does not give any preference for either way ligation. For example, let us imagine an overlap of two segments containing three heterozygous loci. Let the first locus suggest one ligation and the other two loci support the reverse ligation. We can now calculate the turn-error probability between the first and the second locus, but there is no reason to believe that this error has occured on a particular segment of the two. Furthermore, according to the pure turn error model, we have no reason to pick the ligation supported by two loci out of the three. There would be a use for the turn model, if we extended it to capture the observed error variation associated with locus position in the sample. We have earlier noticed that there is higher error probability on the edges of the sequence as described in Figure 4.5, which also suggests an average higher error rate for shorter sequences.

We cannot easily make use of our turn error model, but there is more use of the single locus based model. According to our mixed model, as described in equation (4.7), there is an observed distance-independent probability $\varepsilon_{loc} = 0.012$ that `phase` suggests a wrong phase at some particular heterozygous locus. We can thus use the previously defined confidence calculation formulas (5.1) and (5.2), by just taking

$$C(i, \mathcal{A}) = 1 - \varepsilon_{loc} = 9.988.$$

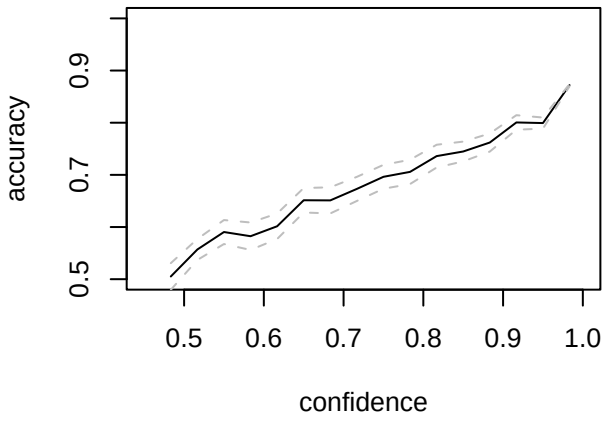


Figure 5.6: Ligation accuracy with respect to confidence of the ligation. There would be some justification to set the confidence threshold near 0.7, but this would not considerably affect the average error rate, as the vast majority of ligations are performed under confidence $0.9 \ldots 1.0$

As long as $\varepsilon_{loc} < 0.5$, the ligation is determined by *majority vote*—picking the ligation phase, which is supported by the majority of loci.

As we are using `phase` as our core infernce tool providing prior locus-specific theoretical probabilities, we believe that the confidence calculation using the locus-specific is more informative than ligation based on the simple majority vote.

Figure 5.6 describes the observed accuracy of ligations, with respect to the calculated confidence. By accuracy we mean the empirically estimated ligation error probability. The plot shows that the calculated confidence is a rough estimator of ligation error—accuracy increases almost linearly with our confidence.

5.2.3 Practical results

We constructed Algorithm 6 to process the overlapped segments of inferred haplotypes. To decide the phase of the segments, we weighted the supporting inferred heterozygotes by the inference confidence estimated by `phase`. We denote the confidence for locus i while inferring segment $\mathcal{A}$ as $0 \leq C(i, \mathcal{A}) \leq 1$.

The algorithm uses the previously defined confidence measures to calculate support for either way ligations. It iterates on heterozygous loci in the overlap to compute the ligation confidence. It then outputs the full ligation confidence allong with ligated segments according to highest support. Computational complexity of Algorithm 6 is linear to the total number of heterozygous loci in the overlaps.

For practical tests we used our HapMap 300-loci datasets from CEU population. Segment inference for the partitioning of the 40 datasets took about 4 hours on 12-node Grid computation cluster. The total number of segments processed was 776. As the long-sequence inference for the same sample takes about 48

Algorithm 6: Ligation using overlaps.

Input: Inferred haplotypes for a genotype $g = h_1 \oplus h_2$ in two consecutive segments $\mathcal{X} = \mathcal{A}_{j-1}\mathcal{A}_j$ and $\mathcal{Y} = \mathcal{A}_j\mathcal{A}_{j+1}$.
Output: Ligation confidence $\mathsf{Conf}_g(\mathcal{X}, \mathcal{Y})$ with ligated genotype $g[\mathcal{X}\mathcal{Y}]$ or an empty set $\emptyset$ if no heterozygous loci are present in the overlap $\mathcal{A}_j$.

function LIGATE($g[\mathcal{X}], g[\mathcal{Y}], j$)
if $|\mathsf{Het}(\mathcal{A}_j)| = 0$ **then return** $\emptyset$
Reset confidences $c_1 = 1.0; c_2 = 1.0$
foreach heterozygous locus $\ell_i \in \mathcal{A}_j$
  $lc = C(i, \mathcal{X})C(i, \mathcal{Y}) + (1 - C(i, \mathcal{X}))(1 - C(i, \mathcal{Y}))$
  if $\mathsf{Hap}_{g[\mathcal{X}]}(i) = \mathsf{Hap}_{g[\mathcal{Y}]}(i)$
    $c_1 = c_1 lc$
    $c_2 = c_2(1 - lc)$
  else
    $c_1 = c_1(1 - lc)$
    $c_2 = c_2 lc$
  end if
end for
$\mathsf{Conf}_g(\mathcal{X}, \mathcal{Y}) = \max\left(\frac{c_1}{c_1+c_2}, \frac{c_2}{c_1+c_2}\right)$
if $c_1 \geq c_2$ **then**
  $g[\mathcal{X}\mathcal{Y}] = h_1[\mathcal{X}]h_1[\mathcal{Y}] \oplus h_2[\mathcal{X}]h_2[\mathcal{Y}]$
else
  $g[\mathcal{X}\mathcal{Y}] = h_1[\mathcal{X}]h_2[\mathcal{Y}] \oplus h_2[\mathcal{X}]h_1[\mathcal{Y}]$
end if
return $\mathsf{Conf}_g(\mathcal{X}, \mathcal{Y}), g[\mathcal{X}\mathcal{Y}]$

end function

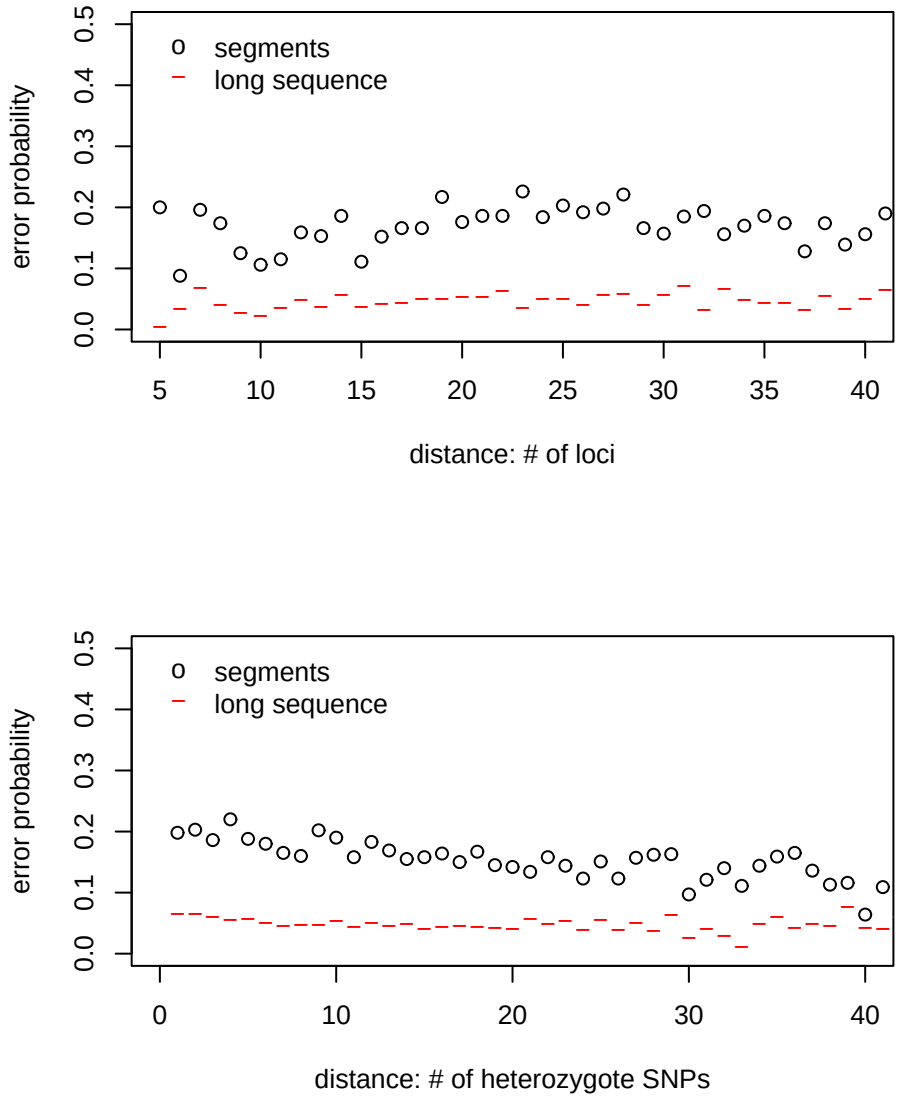


Figure 5.7: Plot compares measured $\mathsf{Lig}(\mathcal{X}, \mathcal{Y})$ error characteristic in segments to the corresponding $\mathsf{Pair}(a, z)$ error rates from inference of unsegmented long sequences. The error rates are plotted by d_{loc} (above) and d_{het} (below). The results were obtained from running our ligation process on HapMap.org CEU population data. Heterozygosity threshold 0.8 was used in partitioning and one heterozygote locus was required in overlaps.

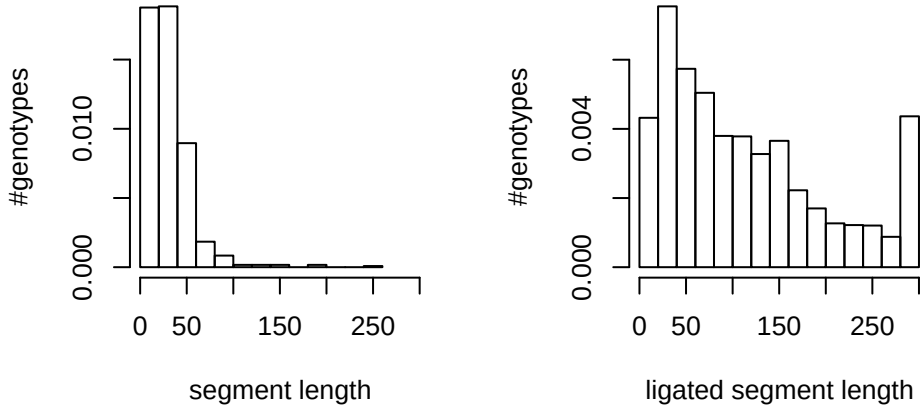


Figure 5.8: The histograms illustrate the length distribution of ligated genotypes in comparison with the lengths of original segments in partitioned samples.

hours on grid (or 25 days on single CPU), there is considerable time efficiency, even though the overlaps double the total length of the sequence. The plots in Figure 5.8 describe segment lengths before and after ligation with overlapping. This illustrates the idea of solving large number of small problems for getting results for larger problems.

From Figure 5.7 we can see that there is a considerable loss of accuracy (3–4 times) in comparison with the long-sequence inference. We plotted the observed pair error rate in long sequences for the same pairs of heterozygotes a and z that were used to calculate $\mathsf{Lig}(\mathcal{X}, \mathcal{Y})$. The results suggest, that the pure confidence based ligation model cannot be directly used in practice, because of its loss of accuracy. The confidence distributions as described in Figure 5.4 show that the vast majority of false ligations were made under a very high confidence support.

Overall error rates in ligated segments are described in Figure 5.9, compared to average error rates as given before in Figure 4.4. This experiment includes all heterozygous pairs in ligated segments. Again, we see considerable loss of accuracy compared to inference in long sequences. Note that the error rates are much higher even at very small distances, which suggests that the cause has not been in the ligation procedure but the inference of underlying segments. It seems, that `phase` is more accurate in inferring haplotypes within long sequences than short ones.

We also investigated the effect of the amount of heterozygosity in the overlaps as described in Figure 5.10, but did not notice strong dependency between the number of heterozygous loci in the overlap and the error probability. For the ligations, which have a high ligation confidence we can believe that `phase` has inferred the overlaps similarly and thus have not made errors on either segments within the overlap. If for an incorrectly ligated overlap $\mathcal{XY} = \mathcal{A}_{j-1}\mathcal{A}_j\mathcal{A}_{j+1}$ the ligation confidence is high, there is reason to believe that phase turn error

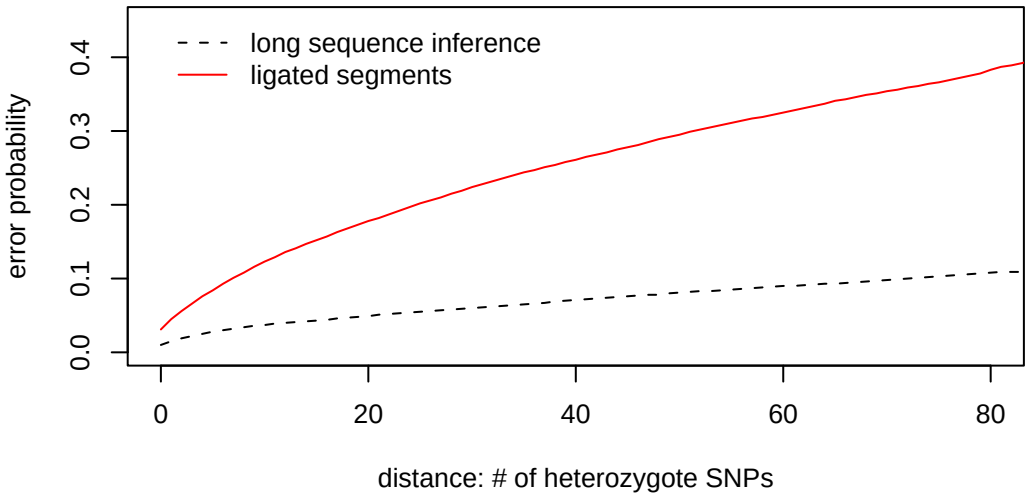


Figure 5.9: Here, the average Pair error characteristic is calculated for all heterozygous pairs in ligated segments. Again, we see considerable loss of accuracy compared to inference in long sequences.

has occured either on $\mathcal{A}_{j-1}$ or $\mathcal{A}_{j+1}$. Appears that `phase` inference model is considerably more accurate for pairs of loci that are not positioned near the edge of the inferred sequence. This is also supported by our initial error measurements described in Figure 4.5.

5.3 Improving ligation accuracy and applicability

Our overlapped ligations work only for such neighbouring segments that have heterozygotes in the overlap. We saw from the heterozygosity analyses that there hardly ever exists a reasonable partitioning where every overlap contains heterozygous loci in the overlap. We therefore need some extra postprocessing to infer phase for such overlaps and possibly also refine the accuracy of overlapped ligations. We propose a couple of theoretical ideas how to approach the task, but do not aim to implement any of these methods within the scope of this work.

By now, we have obtained a partially ligated sample using the heterozygotes in overlaps. This means, that every two consecutive non-homozygous segments have been ligated, but if there exists a homozygous segment somewhere between two segments, the phase has not been determined. For each ligation, we also have a calculated confidence $\mathsf{Conf}_g(\mathcal{A}_j, \mathcal{A}_{j+1})$.

Using our earlier notation, we can describe an example of a partially ligated

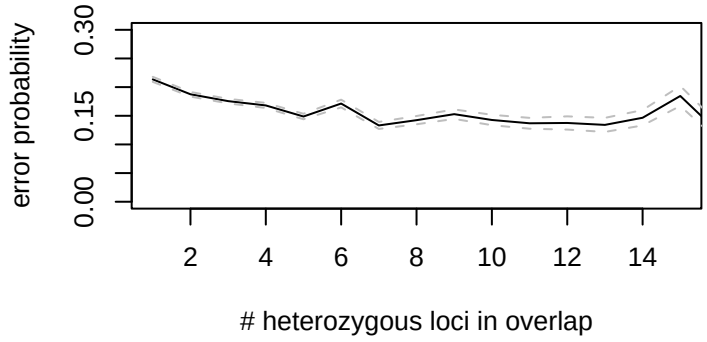


Figure 5.10: Ligation error moderately depends on the number of heterozygous loci in the overlap. We observed only a slight 5% accuracy enhancement when looking at ligations, where the number of heterozygotes is larger that 1.

sample of four genotypes, which has been partitioned into 8 segments.

Hap	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_5$	$\mathcal{A}_6$	$\mathcal{A}_7$	$\mathcal{A}_8$
g_1	1	2	2	2	1	$\perp$	2	2
g_2	2	$\perp$	$\perp$	1	1	1	1	2
g_3	1	1	1	2	2	1	1	1
g_4	2	2	$\perp$	$\perp$	$\perp$	$\perp$	1	2

Having the homozygous region in between, we cannot yet decide, weather the correct phase is

$$\mathsf{Hap}_{g_1}(\mathcal{A}_7\mathcal{A}_8) = 22, \text{ or}$$
$$\mathsf{Hap}_{g_1}(\mathcal{A}_7\mathcal{A}_8) = 11.$$

5.3.1 Markov model approach

This proposed approach nurtures the idea of linkage disequilibrium in its purist form. It assumes that any locus on some sequence is somewhat determined by the sequence preceding the locus. For example, at some position i, we may find that there is a high probability of observing $\ell_i = \mathsf{A}$, when we the sequence preceding the locus is GATG. If the preceding sequence at that position is GTCG, we may equivalently notice a high probability of the sequence be followed by $\ell_i = \mathsf{G}$.

Such correlation of consecutive loci can be formalised as Markov model. For fixed length Markov model, the probability of observing locus i is determined by the previous loci $\ell - 1, \ldots, \ell - d$, i.e. $\Pr[\, h[\ell] \mid h[\ell - 1] \ldots h[\ell - d]\,]$. Hence, we should fix the length d and infer corresponding transition probabilities from successfully phased data. Note that if d is smaller than the length of an intervening homozygotic block, then Markov chain does not help to choose ligation phase, as both haplotypes have equal history and thus equal appearing probabilities. To suggest phase at some heterozygous locus z, the Markov chain must have length

at least equal to the distance from first preceding heterozygous locus a on that genotype.

We can thus estimate the phase of locus z by

$$\Pr\left[\mathsf{Hap}(z) = x | \mathsf{Hap}(a \ldots z-1)\right] \approx \frac{\#\{\mathsf{Hap}(a \ldots z-1)x \in H\}}{\#\{\mathsf{Hap}(a \ldots z-1) \in H\}} \quad .$$

In practice, the algorithm should start by determining the two haplotypes for genotype g in region $a, \ldots, z-1$. These haplotypes differ only in the first locus a. It should then filter out such haplotypes from G, where there has not been any undetermined ligation in $a, \ldots, z-1$ and exactly match one of the haplotypes for g in that region. Then it only needs to calculate the frequencies $\Pr\left[\mathsf{Hap}(z) = 1 \mid \mathsf{Hap}(a) = 1\right]$ and $\Pr\left[\mathsf{Hap}(z) = 2 \mid \mathsf{Hap}(a) = 1\right]$ within the filtered set of haplotypes. When computing the probabilities, we should weigh the observed frequencies by ligation confidences as the sequence $a, \ldots, z$ extends across at least two segment borders. This process can be viewed as training a Markov model on the ligated sample and using it later to estimate the undetermined cases.

This described approach may help to ligate segments separated by homozygotes, but it requires a large sample of genotypes to estimate the conditional frequencies. It is also sensitive to prior mistakes made in segment inference and does not aim to correct any of the prior mistakes. For large samples, it may be possible to extend the considered region to more than one heterozygous locus while mainitaining reasonable statistical significance. This would theoratically enable to bypass some prior errors made on segment edges.

The strength of linkage disequilibrium and locus sampling frequency determine the reasonable length of the Markov model. Refer to section 2.3 for discussion on LD and block structure in haplotypes.

5.3.2 Gibbs sampling

The alternative approach follows the ideas in Gibbs sampling methods, which are the core of haplotype inference in `phase`. Let us recall that in Gibbs iteration, a new haplotype reconstruction for a genotype g is sampled from

$$\{h_u^t, h_v^t\} \leftarrow \Pr\left[H_g \mid G, H_{-g}\right],$$

where H_g refers to the set of valid haplotype configurations for $g \in G$ and H_{-g} denotes the current haplotype reconstruction, but with haplotypes chosen for g being removed.

According to the underlying biological coalescent model, with some probability a suitable haplotype h is chosen from the remaining sample H_{-g} proportional to the frequencies. With residual probability, a new haplotype is made up completely at random. The enhanced sampler version supports recombination, inserting *turns* by some distribution along the chosen haplotype. A complementary haplotype is finally generated for h to make up g.

We can intuitively use the same method for determining the phase of segments rather than individual loci. We can use our calculated ligation confidence to decide

weather or not to insert a turn between a particular pair of consecutive segments. For undetermined ligations, we just take confidence equal to 0.5 for both phases. Furthermore, with some probability we can reject the haplotypes suggested for a genotype in the segment inference step and reconstruct new haplotypes according to the biological model. This would potentially allow to correct some of the mistakes, that were made in the segment inference step.

We do not aim to formulate a consistent sampling model, but give some ideas in the following. Let us look at sampling a pair of haplotypes for g from $\Pr[H_g \mid H_{-g}, G]$, given a partially ligated set of haplotypes over a partitioning $\mathcal{A}_1, \ldots, \mathcal{A}_n$.

Let us start constructing a Markov chain along the partially ligated sequence for g:

1. at a segment junction, with probability p, pick the phase of the next segment proportional to the calculated ligation confidence $\mathsf{Conf}(\mathcal{A}_j\mathcal{A}_{j+1})$. If we have not been able to suggest a ligation by overlapping, take equal confidence for both phases.

2. else with probability $k/(\theta + k)$ pick one of the k other haplotypes for the following segment, which are available in the sample and which suit to constitute g in that segment. Haplotypes are chosen proportional to their observed frequency in the current state of the inferred haploid population. θ is the coalescent population variation parameter as described earlier.

3. else with probability $\theta/(\theta + k)$ construct a new pair of haplotypes for the segment. Randomly choose phase at all heterozygous loci.

4. with a probability q, calculated similar to the original `phase`, add the reconstructed haplotypes to H.

These postprocessing ideas borrowed from Gibbs sampling, are very similar to the haplotype inference implemented in `phase`. The main addition is using the partially ligated reconstruction along with ligation confidences as input. We strongly believe, that Gibbs sampling would be a robust postprocessing method, meriting from the possibility to tune sampling parameters in order to find the best suitable balance between inference accuracy and computational efficiency. Extending `phase` to support such postprocessing would be an important step in finalising the overlapped partitioning method and establishing a robust haplotype inference tool with enhanced computational efficiency.

Conclusions

Haplotype inference has been a challenging and an exciting task in bioinformatics. Recentely there has been even more attentention to the field due to the new availability of extensive samples and associated computational complexity. Statistically minded researchers have provided impressive solutions that seem to be accurate and reasonably efficient for moderate sample sizes.

We have approached the haplotype inference task from the computer science perspective and proposed enhancements to the core algorithms that promise to reduce computational complexity remarkably at some expense regarding accuracy. According to our enhancements, difficult large inference problems are divided into smaller and more tractable ones that can be processed in parallel using high throughput computer farms. In particular, we have proposed a setup that helps to merge solutions from smaller tasks into the final result at a lower computational cost. We have borrowed ideas from shotgun sequencing technique and proposed to use overlaps while partitioning the original sample into segments. Heterozygous loci in the overlaps later suggest the correct way to combine the inferred segments into the full haplotype reconstruction.

We have made an effort in studying the error characteristics of haplotype inference process. We have formulated consistent statistical error models and conducted several practical measurements to study their practicality and to estimate model parameters for specific populations. Although we have used only one inference tool, which is capable of handling reasonably sized samples, our error models are built up to be able to study errors with any inference methods.

Due to the decay of biological linkage disequilibrium between loci as their distance grows, haplotype inference is not reasonable for very long sequences. Our turn error model, for example, allows to assess the theoretical inference error probability for long sequences, having observed the error characteristic on short samples empirically.

We acknowledge that our error models are rather basic in nature and they should be extended to capture further empirically causal factors for inference error. We have already briefly looked at the relations between error rates and error occurence positions on the sample sequence. An interesting follow-up from the error modelling studies would be further research into quantifying the dependencies between observed error rates and estimated population parameters (θ and ρ in particular).

By completing this work we have verified the applicability of our preprocessing enhancement. We have gained support from experiments that this framework potentially provides a valueable computational complexity reduction, when inte-

grated into a robust infernce tool. If this work is to be continued, then the implementation of such a tool or an extension to an existing tool would constitute the next important step forward, providing a novel end-to-end solution for haplotype inference.

Haplotüüpimine kasutades ülekattega segmente

Magistritöö (40AP)

Kristo Käärmann

Sisukokkuvõte

Haplotüüpide tuvastamine genotüpiseerimisandmetest on keerukas bioinformaatika ülesanne. Enamasti on statistilised meetodid ainus praktiline võimalus informatsiooni kogumiseks haplotüüpide tasemel, kuna meetodid haplotüüpide otseseks määramiseks bioloogilisest materjalist puuduvad üldse või osutuvad väga kulukaks. Haplotüüpimiseks on välja pakutud ning tarkvaraliselt implementeeritud mitmeid meetodeid. Olemasolevate vahendite loomisel on eesmärgiks olnud võimalikult täpse haplotüüpideks jaotuse välja pakkumine ning sealjuures on enamasti vaadeldud lühikesi 10–50 SNP pikkuseid sekventse. Pikemate sekventside töötlemine on paljude meetoditega üldsegi võimatu või paremal juhul arvutuslikult ülimahukas.

Antud töös oleme lähenenud haplotüüpide tuvastamisele arvutiteaduslikust vaatenurgast ning pakkunud välja täienduse olemasolevatele meetoditele, mis lubab oluliselt väiksema arvutusressursiga tuvastada haplotüüpe pikkades lõikudes, kaotades samal ajal veidi meetodi täpsuses. Meie väljapakutud täiendus seisneb keeruka arvutusülesande (pika sekventsi) jagamises lühemateks ülekattega segmentideks. Haplotüüpide leidmise lühikestes segmentides usaldame olemasolevatele meetoditele. Lühikeste segmentide töötlemise võib korraldada paralleelselt arvutifarmides ning meie kasutasime sel puhul Eesti Griidi arvutikobaraid. Pika sekventsi kokku kleepimisel kasutame ülekattes esinevaid heterosügootseid lookusi, et valida korrektne kleebe. Sarnast ideed on varem kasutatud näiteks *shotgun* sekveneerimise juures, mis tegi võimalikuks ka inimgenoomi täieliku sekveneerimise.

Magistritöö esimese kolme peatüki jooksul anname ülevaate haplotüüpimise probleemist, peamistest meetoditest ning nende tööpõhimõtetest. Detailsemalt peatume lihtsamatel statistilistel populatsioonimudelitel, mis võimaldavad hinnata haplotüüpide jaotust populatsiooni parameetrite (sh. mutatsiooni ja rekombinatsiooni sagedus) alusel ning vastupidi.

Töö neljandas osas defineerime kaks veamudelit haplotüüpimisel tekkivate vigade kirjeldamiseks. Samuti defineerime veakarakteristikud haplotüüpimise täpsuse hindamiseks empiiriliselt. Praktilisteks katsetusteks koostame testpopulat-

sioonid nii bioloogilistest kui sünteetilistest algandmetest lähtuvalt. Empiiriliste veahinnangute analüüsi tulemusena määrame mõlema veamudeli parameetrid. Defineeritud veamudelid ja -karakteristikud annavad võimaluse hinnata ja võrrelda erinevate haplotüüpimismeetodite täpsust ja omadusi. Põnev edasiarendus siinkirjeldatule oleks veamudelite sidumine populatsiooni (mutatsioon, rekombinatsioon) ja katse (lookuste arv, füüsiline kaugus) parameetritega.

Töö viimane osa on pühendatud haplotüüpimise analüüsimisele ülekattega segmentides. Pakume välja parameetriseeritava algoritmi pika sekventsi ülekattega partitsioneerimiseks ning tõenäosusliku mudeli segmentide kokku kleepimiseks, arvutades sealjuures iga kleepe usaldusväärsuse. Kasutades eelnevalt defineeritud veakarakteristikuid, mõõdame empiiriliselt täpsuse kadu ning kleepimisprotsessi efektiivsust. Vaatleme põgusalt ka potentsiaalseid meetodeid täpsuse parandamiseks Markovi ahelate ja stohhastiliste meetodite abil.

Kokkuvõtteks võime öelda, et oleme töö käigus veendunud ülekattega haplotüüpimise rakendatavuses ning näinud olulist arvutuslikku efektiivsust paralleliseeritavas algoritmi seades. Täpsuse kadu antud protsessis on siiski olnud oodatust kõrgem, seetõttu vajab välja pakutud meetod veel edasiarendust ning analüüsi, et olla praktikas teadlastele kasutatav.

Lisaks juhendajate määravale sisulisele panusele on magistritöö valmimist osaliselt toetanud ETF grant 5722.

Bibliography

[AIJ85] D.J. Aldous, IA Ibragimov, and J. Jacod. *École d'été de probabilités de Saint-Flour XIII-1983*. Springer-Verlag New York, 1985.

[Cla90] Andrew C. Clark. Inference of haplotypes from pcr-amplified samples of diploid populations. *Molecular Biology of Evolution*, 7:111–122, 1990.

[Con05] International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, Oct 2005.

[Cou05] National Research Council. *Mathematics and 21st Century Biology*, chapter 6 - Understanding populations, pages 100–102. National Academies Press, 2005.

[Dem77] Laird N. M. Rubin D. B. Dempster, A. P. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[DS02] Daniel Gianola Daniel Sorensen. *Likelihood, Bayesian and MCMC Methods in Quantitative Genetics*. Springer, 2002.

[EGT04] L. Eronen, F. Geerts, and H. Toivonen. A markov chain approach to reconstruction of long haplotypes. *Pacific Symposium on Biocomputing (PSB 2004)*, 2004.

[EHK03] Eleazar Eskin, Eran Halperin, and Richard Karp. Large scale reconstruction of haplotypes from genotype data. *RECOMB*, 2003.

[Gus02] D. Gusfield. Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. pages 166–175. ACM Press New York, NY, USA, 2002.

[Hap06a] Hapmap.org. Collection of 40 pre-phased datasets from HapMap Central European population. Each dataset contains 300 SNP sample for 89 individuals., 2006.

[Hap06b] Hapmap.org. Collection of 40 pre-phased datasets from HapMap japanese population. Each dataset contains 300 SNP sample for 43 individuals., 2006.

[Hud83] R. R. Hudson. Properties of a neutral allele model with intragenic recombination. *Theor Popul Biol*, 23(2):183–201, Apr 1983.

[Hud02] Richard R Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–8, Feb 2002.

[JSFP+04] Michael I Jensen-Seaman, Terrence S Furey, Bret A Payseur, Yontao Lu, Krishna M Roskin, Chin-Fu Chen, Michael A Thomas, David Haussler, and Howard J Jacob. Comparative recombination rates in the rat, mouse, and human genomes. *Genome Res*, 14(4):528–538, Apr 2004.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[KYF00] M. K. Kuhner, J. Yamato, and J. Felsenstein. Maximum likelihood estimation of recombination rates from population data. *Genetics*, 156(3):1393–401, Nov 2000.

[LM95] L.Excoffier and M.Slatkin. Maximum likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol*, (12), 1995.

[LS03] Na Li and Matthew Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–33, Dec 2003.

[MC05] Gilean A T McVean and Niall J Cardin. Approximating the coalescent with recombination. *Philos Trans R Soc Lond B Biol Sci*, 360(1459):1387–1393, Jul 2005.

[MW06] Paul Marjoram and Jeff D Wall. Fast "coalescent" simulation. *BMC Genet*, 7:16, 2006.

[Nor02] Magnus Nordborg. *Handbook of Statistical Genetics*, chapter Coalescent Theory, Ch 7. Wiley, 2002.

[NQXL02] Tianhua Niu, Zhaohui S Qin, Xiping Xu, and Jun S Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am J Hum Genet*, 70(1):157–169, Jan 2002.

[PBH+01] N. Patil, A. J. Berno, D. A. Hinds, W. A. Barrett, J. M. Doshi, C. R. Hacker, C. R. Kautzer, D. H. Lee, C. Marjoribanks, D. P. McDonough, B. T. Nguyen, M. C. Norris, J. B. Sheehan, N. Shen, D. Stern, R. P. Stokowski, D. J. Thomas, M. O. Trulson, K. R. Vyas, K. A. Frazer, S. P. Fodor, and D. R. Cox. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294(5547):1719–23, Nov 2001.

[Per06] Perlegen dataset. Collection of 5 biological datasets from 21st chromosome by Patil *et. al*. Each dataset contains 300 SNP sample for 30 individuals., 2006.

[SD00] M. Stephens and P. Donnelly. Inference in molecular population genetics. *J. R. Statist. Soc. B*, 62:605–635, 2000.

[SSD01] Matthew Stephens, Nicholas Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *Am. Soc. Human Genetics*, 2001.

[ST98] W.J. Ewens Simon Tavare. *Encyclopedia of Statistical Sciences Update Volume 2*, volume 2, chapter The Ewens Sampling Formula, pages 230–234. Wiley, 1998.

[Syn06] Synthetic data. Collection of 10 synthetic datasets from from simulations using Hudson population simulator. Each dataset contains 300 SNP sample for 89 individuals., 2006.

[Wak06] John Wakeley. *Coalescent Theory: An Introduction*. Roberts & Company, 2006.

[Wea04] Michael E Weale. A survey of current software for haplotype phase inference. *Hum Genomics*, 1(2):141–4, Jan 2004.

[ZCNS02] Kui Zhang, Peter Calabrese, Magnus Nordborg, and Fengzhu Sun. Haplotype block structure and its applications to association studies: power and study designs. *Am J Hum Genet*, 71(6):1386–1394, Dec 2002.