Haplotype Inference through Sequential Monte Carlo

Alexandros Iliadis

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2013

Abstract

Haplotype Inference through Sequential Monte Carlo

Alexandros Iliadis

Technological advances in the last decade have given rise to large Genome Wide Studies which have helped researchers get better insights in the genetic basis of many common diseases. As the number of samples and genome coverage has increased dramatically it is currently typical that individuals are genotyped using high throughput platforms to more than 500,000 Single Nucleotide Polymorphisms.

At the same time theoretical and empirical arguments have been made for the use of haplotypes, i.e. combinations of alleles at multiple loci in individual chromosomes, as opposed to genotypes so the problem of haplotype inference is particularly relevant. Existing haplotyping methods include population based methods, methods for pooled DNA samples and methods for family and pedigree data.

Furthermore, the vast amount of available data pose new challenges for haplotyping algorithms. Candidate methods should scale well to the size of the datasets as the number of loci and the number of individuals are well to the thousands. In addition, as genotyping can be performed routinely, researchers encounter a number of specific new scenarios, which can be seen as hybrid between the population and pedigree inference scenarios and require special care to incorporate

the maximum amount of information.

In this thesis we present a Sequential Monte Carlo framework (TDS) and tailor it to address instances of haplotype inference and frequency estimation problems. Specifically, we first adjust our framework to perform haplotype inference in trio families resulting in a methodology that demonstrates an excellent tradeoff between speed and accuracy. Consequently, we extend our method to handle general nuclear families and demonstrate the gain using our approach as opposed to alternative scenarios. We further address the problem of haplotype inference in pooling data in which we show that our method achieves improved performance over existing approaches in datasets with large number of markers. We finally present a framework to handle the haplotype inference problem in regions of CNV/SNP data. Using our approach we can phase datasets where the ploidy of an individual can vary along the region and each individual can have different breakpoints.

Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to express my deep gratitude to professor Dimitris Anastassiou for his guidance, advice, constant encouragement and trust in my research throughout my study. Deep gratitude is also due to professor Xiaodong Wang who has provided me with constant advice, guidance, numerous suggestions and support. I consider a privilege having professor Anastassiou and professor Wang as my mentors and having worked with them for the past years.

Many special thanks go to my friends Dimitris and Elli Androulakis, Christos Vezyrtzis and Angelika Zavou, with whom I have shared many great moments during the past years.

Finally, I would like to acknowledge the continuing support of my family including my parents Vasiliki and Panagiotis Iliadis and my fiancee Fay Teloni.

Chapter 1

Introduction

1.1 Background

Technological advances in the last decade have given rise to large Genome Wide Studies which have helped researchers get a better inside in the genetic basis of many common diseases. As the number of samples and genome coverage has increased dramatically it is currently typical that individuals are genotyped using high througput platforms to more than 500000 Single Nucleotide Polymorphisms (SNPs).

At the same time theoretical and empirical arguments have been made for the use of haplotypes, i.e. combination of alleles at multiple loci on individual chromosomes, as opposed to genotypes. Population genetic principles show us that variation in populations is inherently structured into haplotypes [1]. The DNA sequence variation that is found in a population is the result of the past transmission of that variation through the population, and this historical past produces a structure to the SNP variation that can be of considerable value in many settings.

This theoretical argument for the use of haplotypes as described above focuses primarily on

the fact that the way genetic variation occurs through mutation, drift and selection recombination, suggests an intrinsic organization into haplotypes. At the same time, it also comes natural to argue that haplotypes represent the natural organization of information within the genome as they also define the functional units of genes [1].

It therefore comes to no surprise that as opposed to examining SNPs independent of each other, haplotypes have been shown to be not only useful, but enabling unique insides in the study of the human genome.

In linkage analysis, haplotype inference can dramatically increase the power over single marker approaches [2]. Haplotypes in a pedigree are invaluable in estimating the identity by descent (IBD) probabilities among pedigree members, which provide the basis of many linkage methods [3]. Haplotype information can also be used to identify genotyping errors, through identification of double recombinations within short chromosomal regions and to infer missing gentoypes.

In association studies a variety of methods exist in the literature that use haplotypes to increase the power of the study to detect causal relationships between a genetic region and a phenotype [4]. Many of these methods use a clustering approach to group haplotypes based on similarity and perform statistical tests on these clusters [5–7]. The underlying concept lies in the expectation that, as noted above, clusters reflect aspects of the evolutionary history of case and control chromosomes. Coalescent based methods more specifically try to explicitly model this history [8–10]. A number of approaches further use a sliding window [5, 11], determined by blocks of strong LD while newer approaches allow localized haplotype clustering that varies along the genome [12].

At the same time haplotypes are required for many population genetic analyses. Specifically, methods for inferring selection [13], for studying recombination [14, 15] as well as historical migration [16, 17] build their subsequent analysis on existing haplotype data.

On the core of all aforementioned methods lies the concept of haplotypes. Experimental methods of haplotyping include diploid to diploid conversion [18], allele specific PCR and cloning. Although these techniques have been shown to provide more information over early developed statistical methods [19,20] they are expensive and extremely time consuming compared to modern high-throughput genotyping.

The computational determination of haplotypes from genotype data is thus potentially very valuable if the estimation can be done accurately and has received an increasing amount of attention over recent years. Existing haplotyping methods include population based methods, methods for pooled DNA samples and methods for family and pedigree data.

In the current era of genomewide association studies the vast amount of available data pose new challenges towards this endeavor. Candidate methods should scale well to the size of the datasets as the number of loci and the number of individuals are well to the thousands. At the same time, as genotyping can be performed routinely and at a low cost, researchers encounter a number of specific new scenarios which can be seen as hybrid between the population and pedigree inference scenarios that require special care to incorporate the maximum amount of information.

In this thesis we develop a Sequential Monte Carlo framework to address these issues. Our methodology examines all alternative phasing scenarios locally in a tree-like fashion and thus the name of our developed model ”Tree based Deterministic Sampling” (TDS).

1.2 Overview of Prior Work on Haplotype Inference

As haplotype analysis offers advantages over genotypes it is not surprising that this subject has received a lot of attention. Haplotype inference or ”phasing” refers to the reconstruction of the un-

known true haplotype configuration from observed genotype data. There are two main settings for haplotype inference as mentioned in the previous paragraph, namely inference in population samples and inference in pedigrees. Both settings suffer from the same problem, that the space of all consistent haplotype configurations for each individual in the dataset is intractable. A substantial amount of prior work has been done for both phasing scenarios.

In pedigrees a large number of statistical and genetic rule based methods were developed to estimate the true haplotype configuration by identifying a single most likely consistent haplotype configuration [3]. Haplotyping methods for pedigrees include likelihood-based methods and genetic rule-based methods. Likelihood-based methods reconstruct configurations by maximizing the likelihoods or conditional probabilities of the configurations. Rule based algorithms reconstruct configurations by minimizing the total number of recombinants in the pedigree data.

In unrelated individuals a number of methodologies have been developed as well, that use appropriate statistical frameworks to determine for each individuals its most probable haplotype orientation [21]. Population based methods attempt to take advantage and efficiently process the inherent haplotype information in the population samples to perform the inference. No rule based approaches can be applied in this setting as familial information for the samples is absent.

In addition, in recent years hybrid scenarios between these two haplotype inference categories have come to existence with trio families (families consisting of father-mother-offspring) being the most prevalent. A number of algorithms originally developed for haplotype inference in unrelated individuals have been extended to the trio case.

In the following chapters existing approaches for both settings will be described in detail. The limitations for each category of methods will be explicitly stressed together with their potential advantages as well as weaknesses.

1.3 Contributions of the Thesis

In Chapter 2, a Sequential Monte Carlo methodology for haplotype inference in trio datasets is introduced and its performance is demonstrated against competing methods. The performance and scaling of the method is demonstrated in large datasets which is the case in current Genome Wide Association Studies (GWAS).

In Chapter 3, the previously developed trio framework is extended to handle cases in which families may include more than one siblings. In the extended framework, we attempt to resolve the phasing within families using a modified pedigree rule-based approach and resort to the population derived information for ambiguous phasing instances. We demonstrate and quantify the gain in the accuracy offered by applying the proposed methodology as opposed to any of the conventional trio or pedigreed configurations that could partially resolve the same problem.

In Chapter 4, a framework for the case of pooling and polyploid data is presented. The framework can process larger genotype segments compared with current approaches while maitaining the accuracy for smaller segments.

In Chapter 5, a related new framework is introduced for inferring phase in regions including simultaneously Copy Number Variations (CNVs) as well as SNPs. As opposed to pooling or polyploid methods that have to assume fixed ploidy across the phasing region, we address the general problem where the ploidy of each individual can vary across the region and each individual can have different breakpoints.

In the last Chapter conclusions and future work are discussed.

Chapter 2

Haplotype Inference in Trio Families

2.1 Introduction

As discussed in the previous chapter a very common setting in association studies is the case of ”trio” data, HapMap [22] being an example. In particular, ”trio” data consist of genotypes given in father-mother-child triplets and are widely obtained in GWAS. Most phasing algorithms originally developed for unrelated individuals have been adapted to this type of data. A major challenge currently for all previously developed and new frameworks, as will be further discussed along this and the following chapters, is to be able to scale well both accuracy and computationally wise with the constantly increasing marker and dataset sizes.

One of the most common algorithms for trio inference which was also used in the HapMap project is PHASE [23]. PHASE uses a Bayesian approach attempting to capture the tendency that haplotypes cluster together over regions of the chromosome and that this clustering can change as we move along the chromosome because of recombination. It uses a flexible model for the decay of linkage disequilibrium (LD, the non-random association of alleles) with distance. Al-

though PHASE is considered the most accurate method, its computational complexity makes it prohibitively slow even for intermediate-sized datasets. Thus, it may not be the method of choice for routine use in large genome-wide association studies. On the other extreme of the trade-off between complexity and accuracy, a computationally simple method (2SNP [24]) uses maximum spanning trees to successively phase whole genotypes starting from SNP pairs. Other well known approaches include HAP [25], using imperfect phylogeny, HAP2 using a Markov Chain Monte Carlo (MCMC) scheme [26] and PL-EM [27], which uses an Expectation Maximization (EM) algorithm. A Gibbs sampling method, Haplotyper, is proposed in [28], which introduces the partition-ligation (PL) method to support haplotype inference on long genotype vectors, a procedure adopted by some of the aforementioned methods so that they can be extended to large datasets. An obvious drawback of the Gibbs sampler and of most of the previous frameworks is that when new data is introduced into the original dataset, the previous data also has to be reused in the estimation of the new data. Another drawback of using Gibbs sampler and EM algorithm in the haplotype inference problem is the lack of robustness of these two algorithms when the parameter space exhibits multimodality such as the one we encounter in the haplotype inference problem. The performance of these methods has been evaluated in simulated datasets of both trio as well as unrelated individuals in a comparative review [21], providing some ”gold standard” datasets for future algorithms to be compared upon. A more recent approach (BEAGLE [29, 30]) uses localized haplotype clustering and fits the data using an EM-style update.

As noted above, it is important for phasing methods that they scale well with the number of SNPs as well as the number of individuals. It is also important in terms of computational time that when new data is inserted in phased datasets, we do not have to reuse the previous data in the estimation of the new data.

In this chapter we introduce our TDS algorithm for haplotype phasing of trio data. In our methods trios are processed sequentially. All possible solutions for each haplotype are examined. TDS uses the idea that within haplotype blocks there is limited haplotype diversity and thus attempts to phase each new trio using haplotypes that have already been encountered in the previously seen trios. The TDS framework allows us to effectively perform this search in the space of all possible solution combinations. The procedure as will be described in the ”Methods” section that follows, can be seen as an efficient tree search procedure where in each step only ”the most probable” solution streams are kept. Each of them contains one and only one solution for each trio already encountered.

In the following subsections we present our algorithm and we show that TDS demonstrates an excellent tradeoff of accuracy and speed, making it particularly suitable for routine use.

2.2 Results

The structure of this section is as follows: First we describe the datasets and figures of merit used to evaluate the method. Then we present the results from comparing our method to BEAGLE, PHASE and 2SNP.

2.2.1 Datasets

We used a set of simulated datasets with the ”COSI” software [31] as provided in [21]. The haplotypes were simulated using a coalescent model that incorporates variation in recombination rates and demographic events and the parameters of the model were chosen to match aspects of data from a sample of Caucasian Americans [21, 31]. Three classes of dataset were provided, with

each consisting of 20 sets of 30 trios spanning 1 Mb of sequence with a density of 1 SNP per 5 kb [21] .

We also used the ”COSI” software to create our own realistic simulated data sets to assess the performance of our method on large datasets. We created 20 datasets each of them consisting of 4000 haplotypes with 20 Mb of marker data using the ”best-fit” parameters obtained from fitting a coalescent model to the real data. Samples were taken from a European population and each simulated dataset has a recombination rate sampled from a distribution matching the deCODE map [32], with recombination clustered into hotspots. For each simulated dataset, we intitially selected only those markers with minor allele frequency greater than 0.05. Markers were then randomly selected to obtain a density of about 1 SNP per 3kb. In each dataset two sample sizes were created: 100 and 1000 trios. In each trio, each parent was randomly assigned a haplotype from the population so that no two individuals had the same haplotype and one of the haplotypes of each parent was selected to be transmitted to the child.

2.2.2 Definitions of Criteria

Transmission Error Rate: The transmission error rate is the proportion of non-missing parental genotypes with ambiguous phase that were incorrectly phased [30].

Incorrect Trios (IT): The number of trios for which phasing was not completely correct.

Computational Time: Our algorithm was implemented in Java for portability, memory efficiency and speed. For each method we recorded the average computational time in each dataset (ST1, ST2, ST3) on a 3.66 GHz Xeon Intel PC with 8 GB of RAM.

2.2.3 Transmission Error Rate and Incorrect Trios

The performance of the methods on the simulated data sets is shown in Tables 2.1 and 2.2 . We decreased the <nsamples> parameter in BEAGLE from the default value, $R = 4$, to decrease computational time. Our purpose was to make the results of BEAGLE and TDS as comparable as possible by allowing both methods to run for approximately the same time. PHASE shows superior performance to all other methods in all datasets for both figures of merit. 2SNP was consistently outperformed by all other methods consistent with the result mentioned in [29]. For most of the datasets, a lower transmission error rate usually implied fewer incorrectly-phased individuals. TDS shows superior performance to BEAGLE and 2SNP for all datasets, losing only to PHASE.

We set 1% of the genotypes to missing values and we reevaluated the performance of the algorithm in these datasets and the results are shown in Tables 2.3 and 2.4. We again see that TDS shows superior performance compared to BEAGLE with <nsamples> parameter equal to 1 on all datasets. When we set in BEAGLE <nsamples>=4, BEAGLE shows superior performance on ST3 dataset and marginally on ST1 dataset.

We demonstrated the accuracy of our method with increasing dataset size by varying the number of trios and markers and evaluated the performance by means of the Transmission Error Rate as shown in Table 2.5. We used marker sizes of 200, 400, 1000 and 6000 markers for 100 and 1000 trios. Due to the excessive computational time of PHASE, we excluded it from these comparisons. Furthermore, we avoided using the number of Incorrect Trios as means of comparison, because as the genotype vectors grow longer, eventually all methods will find it hard to correctly infer the entire haplotype and the number of Incorrect Trios will be the total number of trios. For datasets of the size of 1000 trios we noted that, in order to be able to take advantage of the information offered

Average Transmission Error Rate			
	ST1	ST2	ST3
PHASE	0.0013	0.0013	0.0145
BEAGLE			
R=1	0.0235	0.0318	0.0426
R=4	0.0150	0.0148	0.0344
TDS	0.0039	0.0065	0.0320
2SNP	0.4377	0.4868	0.4861

Table 2.1: Average transmission Error Rate For Phasing Trios.

Average number of Incorrect Trios per dataset			
	ST1	ST2	ST3
PHASE	0.3	0.4	2.45
BEAGLE			
R=1	3.75	5.8	6.4
R=4	1.95	2.9	5.45
TDS	0.95	1.6	5.4
2SNP	25.9	28.6	28

Table 2.2: Average number of Incorrect Trios per dataset

as a whole, we had to allow a very large number of streams in our algorithm (Methods section) that would result in excessive computational time. However, we found that we could have minor losses by partitioning the dataset in slices of 100 trios where we had established significant gain compared to BEAGLE. From Table 2.5 we see that TDS shows superior performance for datasets of up to 100 trios for all marker sizes. For datasets of the size of 1000 trios, BEAGLE showed superior performance to all methods.

Average Transmission Error Rate			
	ST1	ST2	ST3
PHASE	0.0031	0.0023	0.0161
BEAGLE			
R=1	0.0213	0.0248	0.0354
R=4	0.0093	0.0133	0.0278
TDS	0.0094	0.0116	0.0348
2SNP	0.3038	0.3486	0.3169

Table 2.3: Average transmission Error Rate For Phasing Trios with 1% Missing Rate.

Average number of Incorrect Trios per dataset			
	ST1	ST2	ST3
PHASE	0.6	0.475	2.653
BEAGLE			
R=1	3.6054	5.25	6.4661
R=4	1.7464	3.1321	4.8893
TDS	1.7521	2.7018	5.7768
2SNP	26.05	28.55	28.2

Table 2.4: Average number of Incorrect Trios per dataset with 1% Missing Rate

		Markers			
		200	400	1000	6000
TDS	100	0.00063	0.00075	0.0015	0.0023
	1000	0.00042	0.0008	0.0015	0.0023
BEAGLE	100	0.00013	0.0013	0.0021	0.0024
	1000	0.000011	0.00033	0.0005	0.0007
2SNP	100	0.1094	0.2855	0.3916	0.4315
	1000	0.1733	0.2524	0.3836	0.4117

Table 2.5: Average Transmission error rate for 100 and 1000 Trios as a function of the number of markers

2.2.4 Timing Results

The computational times for datasets ST1, ST2 and ST3 are displayed in Table 2.6. In Table 2.7 we present the average running time on the same datasets, but with randomly inserting 1% missing SNPs in each one of them. Based on these times 2SNP is the fastest algorithm followed by TDS. Both algorithms were faster than the fastest BEAGLE runs done with <nsamples> parameter equal to 1. PHASE was the slowest algorithm with computational times 3 orders of magnitude more than the remaining three algorithms. In Table 2.8 we demonstrate that for large datasets TDS scaled almost linearly with the number of markers and, as described in the previous subsection, with the number of trios. For datasets of up to 100 trios, our method is faster than BEAGLE; however for datasets of 1000 trios, BEAGLE is the fastest of all methods for marker sizes up to 400 markers.

	Time (s)		
	ST1	ST2	ST3
PHASE	8452	4932	5464
BEAGLE			
R=1	2.59	2.73	2.95
R=4	2.8	3.18	3.27
TDS	1.99	2.48	2.61
2SNP	0.63	0.6	0.59

Table 2.6: Timing Results

	Time (s)		
	ST1	ST2	ST3
PHASE	8613	5220	5831
BEAGLE			
R=1	2.6744	2.9873	3.2409
R=4	2.9233	3.2858	3.4429
TDS	2.0643	2.5815	2.7484
2SNP	0.67	0.63	0.6

Table 2.7: Timing Results with 1% Missing Rate

		Markers			
		200	400	1000	6000
TDS	100	2.8	5	14.4	113.6
	1000	31.8	63.3	156.2	1257.4
BEAGLE	100	3.7	5.6	15.2	118.4
	1000	12.7	31.6	291.8	1952.4
2SNP	100	3	8.9	28.7	180.7
	1000	33.4	116.2	399.8	3008.2

Table 2.8: Average Timing Results in seconds for 100 and 1000 Trios as a function of the number of markers

2.2.5 Memory requirements

All methods could complete the experiments within the preallocated 1.5 Gb of RAM.

2.3 Discussion

An important feature of our algorithm is the partition of the whole genotype sequence in smaller blocks that exhibit limited haplotype diversity. We currently identify these haplotype blocks based on the genotype sequences (see Haplotype Block Partitioning section). However, we can have significant gain in the accuracy of our algorithm if we improve the accuracy in the estimation of the boundaries of the haplotype blocks. To achieve that, either the haplotype blocks should be already known from outside sources, or a set of phased haplotypes from the region at interest should be already available. In real applications, it is very often the case that studies are performed in populations that are already studied in the HapMap project. This means that for these populations we have accurately phased samples, which can be used as basis for accurate definition of the haplotype blocks. Our methodology offers a unique framework that can easily incorporate prior knowledge in the form of haplotypes or trio genotypes from the same population as that from which the target samples were drawn. In the case of haplotypes (such as those available from the HapMap), they are introduced in the form of a prior for the counts in the TDS algorithm. In the case of unphased trio genotypes, the trios can be phased along with the target samples, with the result discarded at the end. The presence of the extra information will improve the phasing accuracy of the target samples.

2.4 Methods

2.4.1 Brief Description

We first give an intuitive description of our algorithm highlighting its major concepts without going into detailed mathematical formalization. Suppose that we denote the wild allele in a particular SNP locus in a haplotype as ”0” and the rare allele as ”1”. Similarly in a genotype we denote with 0 that the individual is homozygous to the wild allele at that SNP and with ”1” that the individual is homozygous to the rare allele. We denote with ”2” the heterozygous case. For example the haplotype pair ”10110” and ”00100” would produce the genotype ”20120”.

In nuclear families each parent transmits a chromosome to a child. In most cases we can detect which parent transmitted which SNP to the offspring based on the genotypes of the parents and the offspring. The only case where we cannot infer that information is when both parents and the offspring are all heterozygous to that SNP (i.e., at that SNP all three genotypes are ”2”). In that case either parent can have transmitted the wild or the rare allele so we have two possibilities for the origin of each allele. This means that if a genotype of a trio has L ambiguous SNPS then this trio would have 2^L possible solutions (see solutions for the trios in Figure 2.1).

Our algorithm processes nuclear families sequentially (Figure 2.1). In each family multiple solutions are produced when we encounter a triple heterozygote SNP as explained before. Our algorithm examines all these different possible solutions.

Suppose we had n trios and each one of them had $\{K_1, ..., K_n\}$ possible solutions. If we evaluate simultaneously all solutions for all haplotypes, which would obviously be the optimal way, we would end up with a total of $\prod_{i=1}^{n} K_i$ possible solutions each one of them having one and only one solution for each trio. To be consistent we will call ”solution” only the final solution and we will

call these potential solutions as solution streams. Clearly this number of solution streams would be infeasible for all non trivial applications. Instead, in our algorithm we process trios sequentially and after processing each trio we keep only a pre-specified K number of solution streams that would be the most probable ones (Figure 2.1, 2b and 3b keeping only $K = 2$ streams in the end of these steps). Each one of these streams would have one and only one solution for each trio we have encountered (Figure 2.1).

To further explain this procedure suppose that after processing a trio we have K streams. When the next trio is processed, which has, say, K^{ext} possible solutions, we append each of these solutions to each of the previous K streams resulting in a total of KxK^{ext} streams (Figure 2.1, 2a and 3a). From these streams we keep only the K most probable ones (Figure 2.1, 2b and 3b). So we always end up with K streams after processing each trio.

The idea for weighting the different streams is based on the concept that within a haplotype block we expect to have limited diversity and find only a subset of all the possible haplotypes. This means that most haplotypes should be encountered more than once. In terms of our procedure we would like to phase each new trio based on haplotypes that we have already encountered in that stream. Since the weight we assign to each node should capture this feature, it is a function of the weight that this node had prior to attaching one of the possible solutions of the current trio and of a factor that represents how the currently appended solution includes haplotypes that have already been seen (see Eq. 2.5 in Methods section).

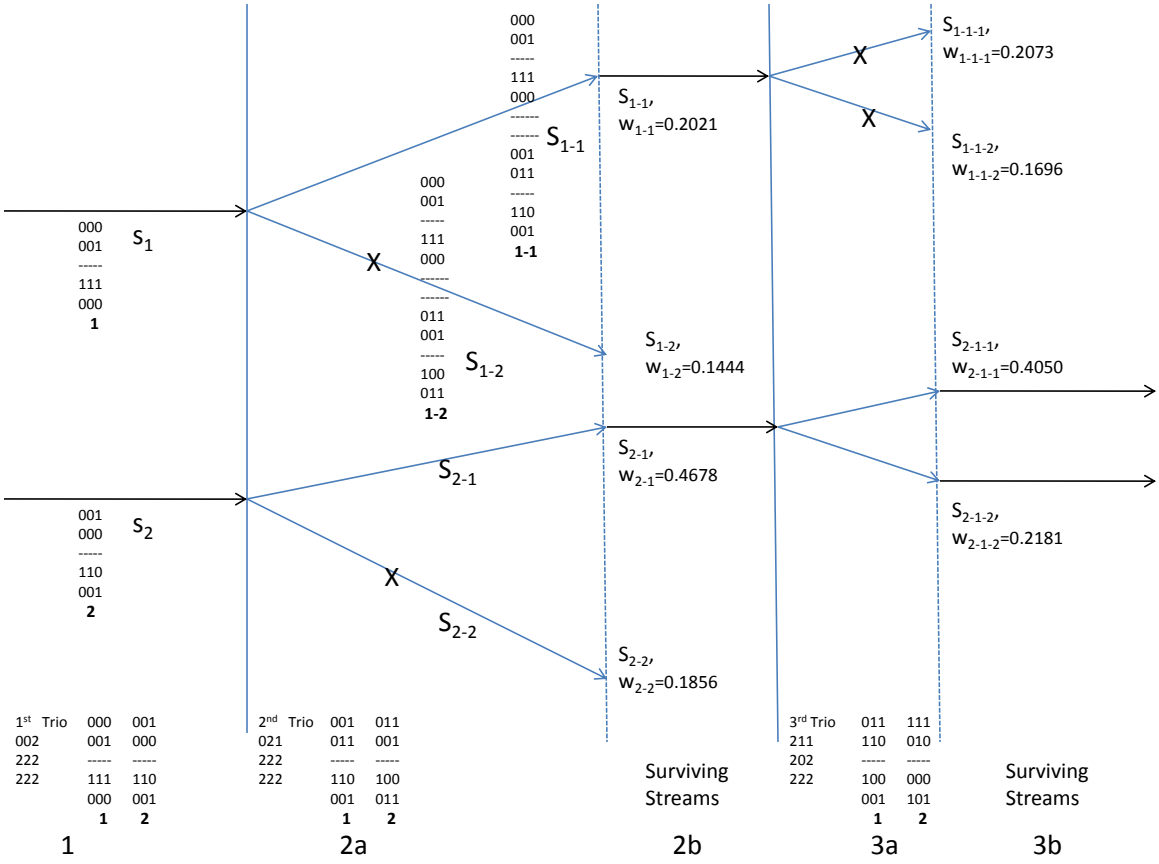

Figure 2.1: Example of TDS. We process three trios sequentially. In each trio the first two genotypes are the genotypes of the parents and the third genotype is the genotype of the child. The possible solutions of each trio are given exactly next to it and numbered 1, 2. In each of the possible solutions for each trio the first two genotypes are the transmitted and the untransmitted haplotype from the first parent and similarly the remaining two for the second parent. At each step we are willing to keep only $K = 2$ streams which would be called surviving streams. 1) The first trio has two possible solutions. 2) a) The second trio has two possible solutions. We have four possible combinations of a solution from the first trio to a solution from the second. The indices below the solutions show from which solutions from each trio this stream was created. For example stream s_{1-2} as illustrated, was created from the first solution in the first trio and from the second in the second. In each stream we associate a weight as described in method section. b) We keep only the $K = 2$ streams with the highest weights (surviving streams) so at this point we consider them as the most probable and keep them. 3) The third trio has 2 possible solutions. a) Each one of them is appended in the end of each of the two solutions that we have kept. The definition of the streams is similar as before with stream s_{2-1-1} coming from appending solution 1 of the third trio to stream s_{2-1}. b) Again we keep only two of the streams the ones with the highest weights s_{2-1-1} and s_{2-1-2}.

2.4.2 Definitions and Model Selection

Let us assume that we have N trios genotyped in L SNPS. Suppose that g_t are the genotypes of the t^{th} trio, i.e., $g_t = \{g_{t,f}, g_{t,m}, g_{t,c}\}$ where $g_{t,f}$, $g_{t,m}$, $g_{t,c}$ are the genotypes of the father the mother and the child of trio t respectively. Suppose also that $G_t = \{g_1, ..., g_t\}$ is a set of genotypes of trios up to and including trio t. In each trio we consider the haplotypes of the parents denoted as $h_t = \{h_{t,1}, h_{t,2}, h_{t,3}, h_{t,4}\}$, where $\{h_{t,1}, h_{t,2}\}$ are the two haplotypes of the first parent and $\{h_{t,3}, h_{t,4}\}$ are the two haplotypes of the second parent and similarly define $H_t = \{h_1, ..., h_t\}$. Let us also define as $\theta = \{\theta_1, ..., \theta_M\}$ a set of population haplotype frequencies for all the M haplotypes that appear in the population and $Z = \{Z_1, ..., Z_y\}$ as the set of haplotypes compatible with at least a genotype of any trio.

In the next subsection, we present the form that the TDS Estimator would have should the haplotype frequencies were known. Then we move forward and make the connection to the real scenario were the haplotype frequencies (θ) are not known.

2.4.3 TDS Estimator with known population haplotype frequencies θ

Similarly to Sequential Monte Carlo methods we assume that by the time we have processed genotype g_{t-1} we have a set of solution streams and their associated weights $\{H_{t-1}^{(k)} | w_{t-1}^{(k)}, k = 1, \ldots, K\}$ properly weighted with respect to their posterior distribution $p_\theta(H_{t-1}|G_{t-1})$. When we process the trio t we would like to make an online inference of the haplotypes H_t based on the genotypes G_t. From Bayes' theorem we have

$$\begin{aligned} p_\theta(H_t|G_t) &\propto p_\theta(g_t|H_t, G_{t-1}) p_\theta(H_t|G_{t-1}) \\ &\propto p_\theta(g_t|H_t, G_{t-1}) p_\theta(h_t|H_{t-1}, G_{t-1}) p_\theta(H_{t-1}|G_{t-1}) \end{aligned} \tag{2.1}$$

Given the set of solution streams and the associated weights we approximate the distribution $p_\theta(H_{t-1}|G_{t-1})$ as follows:

$$\hat{p}_\theta(H_{t-1}|G_{t-1}) = \frac{1}{w_{t-1}} \sum_{k=1}^{K} w_{t-1}^{(k)} I(H_{t-1} - H_{t-1}^{(k)})$$

where $w_{t-1} = \sum_{k=1}^{K} w_{t-1}^{(k)}$ and $I(*)$ is the indicator function such that $I(x-y) = 1$ for $x = y$ and $I(x-y) = 0$ otherwise.

From the previous relationships, if we knew the system parameters θ, and assuming that there are K^{ext} possible haplotypes compatible with the genotype of the t^{th} trio, we would be able to approximate the posterior distribution of $p_\theta(H_t|G_t)$ as:

$$\hat{p}_\theta(H_t|G_t) = \frac{1}{w_t^{ext}} \sum_{k=1}^{K} \sum_{i=1}^{K^{ext}} w_t^{(k,i)} I(H_t - [H_{t-1}^{(k)}, h_t^{(i)}]) \tag{2.2}$$

where $[H_{t-1}^{(k)}, h_t^{(i)}]$ represents the vector obtained by appending the element $h_t^{(i)}$ to the vector $H_{t-1}^{(k)}$ and $w_t^{ext} = \sum_{i,k} w_t^{(k,i)}$ with

$$w_t^{(k,i)} \propto w_{t-1}^{(k)} p_\theta(g_t|h_t = i) p_\theta(h_t = i|H_{t-1}^{(k)})$$

2.4.4 TDS Estimator with unknown population haplotype frequencies θ

However, the population haplotype frequencies are not known. Suppose now that their posterior distribution H_t and G_t only depends on a set of sufficient statistics $T_t = T_t(H_t|G_t) = T_t(T_{t-1}, h_t, g_t)$.

Similar to 2.1 we have:

$$\begin{aligned} p_\theta(H_t|G_t) &\propto p_\theta(g_t|H_t, G_{t-1}) p_\theta(h_t|H_{t-1}, G_{t-1}) p_\theta(H_{t-1}|G_{t-1}, Z) \\ &\propto p_\theta(H_{t-1}|G_{t-1}, Z) p_\theta(g_t|H_t, G_{t-1}) \int p(h_t|H_{t-1}, \theta, Z) p(\theta|T_{t-1}, Z) d\theta \\ &\propto p_\theta(H_{t-1}|G_{t-1}, Z) \int p(h_t|H_{t-1}, \theta, Z) p(\theta|T_{t-1}, Z) d\theta \end{aligned} \tag{2.3}$$

as we only consider haplotypes that are compatible with the genotype of the t^{th} individual. The recursion now lies only in computing the integral in 2.3. In order to calculate the integral in the previous equation we will define the prior distribution for the parameters θ and we will show how to update their posterior distribution.

2.4.5 Prior and Posterior Distribution for θ

Assuming random mating in the population it is clear that the number of each unique haplotype in H is drawn from a multinomial distribution based on the haplotype frequency vector θ [33]. This leads us to the use of the Dirichlet distribution as the prior distribution for θ so that $\theta \sim D(\rho_1, \ldots, \rho_M)$. It is well known in Bayesian statistics that the Dirichlet distribution is the conjugate prior of the multinomial distribution. This implies in our case that if we assume that the prior distribution for θ is Dirichlet and we draw haplotypes based on their frequencies (multinomial distribution), then the posterior distribution for θ is again a Dirichlet distribution. We prove this fact below:

$$
\begin{aligned}
& p(\theta|G_t, H_t, Z) \\
& \propto p(g_t|h_t = (h_{t,1}, h_{t,2}, h_{t,3}, h_{t,4}), \theta, G_{t-1}, H_{t-1}) \\
& xp(h_t = (h_{t,1}, h_{t,2}, h_{t,3}, h_{t,4})|\theta, G_{t-1}, H_{t-1}) p(\theta|G_{t-1}, H_{t-1}) \\
& \propto p(h_t = (h_{t,1}, h_{t,2}, h_{t,3}, h_{t,4})|\theta, Z) p(\theta|G_{t-1}, H_{t-1}, Z) \\
& \propto \theta_{h_{t,1}} \theta_{h_{t,2}} \theta_{h_{t,3}} \theta_{h_{t,4}} \prod_{m=1}^{M} \theta_m^{\rho_m(t-1)-1} \\
& \propto \prod_{m=1}^{M} \theta_m^{\rho_m(t-1)-1+\sum_{i=1}^{4} I(z_m - h_{t,i})} \\
& \propto D(\rho_1(t-1) + \sum_{i=1}^{4} I(z_1 - h_{t,i}), \ldots, \rho_M(t-1) + \sum_{i=1}^{4} I(z_M - h_{t,i}))
\end{aligned}
$$

where we denote $\rho_m(t)$ $m = 1, \ldots, M$ as the parameters of the distribution of θ after the t^{th} trio and $I(z_m - h_{t,i})$ with $i = 1, \ldots, 4$ is the indicator function which equals 1 when $z_m - h_{t,i}$ is a vector of zeros, and 0 otherwise.

2.4.6 TDS Estimator

Taking into consideration as argued above that if we know the systems parameters θ then the $p(h_t|H_{t-1}, \theta, Z)$ term represents sampling from a multinomial distribution and that the mean of the Dirichlet distribution with respect to an element θ_k of the vector θ is

$$E\{\theta_k\} = \frac{\rho_k}{\sum_{j=1}^{M} \rho_j}$$

we calculate the integral in 2.3 as follows:

$$\begin{aligned}
&\int p(h_t = (h_{t,1}, h_{t,2}, h_{t,3}, h_{t,4}) | H_{t-1}, \theta, Z) p(\theta | T_{t-1}, Z) d\theta \\
&= E_{\theta|T_{t-1}} \{ \theta_1^{\sum_{i=1}^4 I(z_1 - h_{t,i})} \dots \theta_M^{\sum_{i=1}^4 I(z_M - h_{t,i})} \} \\
&= \int \theta_1^{\sum_{i=1}^4 I(z_1 - h_{t,i})} \dots \theta_M^{\sum_{i=1}^4 I(z_M - h_{t,i})} \frac{1}{B(\rho(t-1))} \prod \theta_i^{\rho_i(t-1)-1} d\theta \\
&= \frac{B(\rho(t-1)+r)}{B(\rho(t-1))} (\int \frac{1}{B(\rho(t-1)+r)} \prod \theta_i^{\rho_i(t-1)+\sum_{i=1}^4 I(z_M - h_{t,i})-1} d\theta) \\
&= \frac{B(\rho(t-1)+r)}{B(\rho(t-1))}
\end{aligned}$$

where $r = [\sum_{i=1}^4 I(z_1 - h_{t,i}), \dots, \sum_{i=1}^4 I(z_M - h_{t,i})]$ and $B(\rho(t-1)) = \frac{\prod_{i=1}^M \Gamma(\rho_i(t-1))}{\Gamma(\sum_{i=1}^M \rho_i(t-1))}$

Having calculated the integral, we can go back to the recursion and assuming that we have approximated $p(H_{t-1}|G_{t-1})$, we can approximate $p(H_t|G_t)$ as

$$\hat{p}_\theta(H_t|G_t) = \frac{1}{w_t^{ext}} \sum_{k=1}^{K} \sum_{i=1}^{K^{ext}} w_t^{(k,i)} I(H_t - [H_{t-1}^{(k)}, h_{t,1}^i, h_{t,2}^i, h_{t,3}^i, h_{t,4}^i]) \tag{2.4}$$

The weight update formula is given by

$$w_t^{(k,j)} \propto w_{t-1}^{(k)} \frac{B(\rho^{(k)}(t-1)+r)}{B(\rho^{(k)}(t-1))} \tag{2.5}$$

where as noted above $r = [\sum_{i=1}^4 I(z_1 - h_{t,i}^j), \dots, \sum_{i=1}^4 I(z_M - h_{t,i}^j)]$ and $\rho^{(k)}(t-1)$ is the parameter vector of the assumed Dirichlet prior which represents how many times we have encountered each haplotype in stream k in the solutions up to family $t-1$.

2.4.7 Haplotype Block Partitioning

Again we use the idea that haplotypes exhibit block structures so that within each block the haplotype blocks exhibit limited diversity compared to the whole haplotype vectors. To define these blocks we use a Dynamic Programming (DP) algorithm similar to the one used in [34] so that we partition G into subsets of genotype segments. Our criterion for the DP algorithm partition would be that the sum of the entropies of the genotypes of the individual blocks would be minimum.

Let us define C(j) as the minimum total block entropy up to the j^{th} SNP, where total block entropy is the sum of the entropies of all the blocks. If $G_{i:j}$ is the set of genotypes that contains genotype segments from SNP i to SNP j, the entropy $E(i,j)$ of that segment can be computed from the number of occurences of each unique genotype segment in $G_{i:j}$.

More specifically if there are n distinct genotypes in $G_{i:j}$, $\{g_1, g_2, \ldots, g_n\}$ each one of them with counts $\{a_1, a_2, \ldots, a_n\}$ then $E(i,j) = -\sum_{k=1}^{n} p_k ln(p_k)$, where $p_k = \frac{a_k}{\sum_{i=1}^{n} a_i}$. The DP algorithm then can be formulated as the following recursive structure:

$$C(j) = min_{1 \leq i \leq j}\{C(i-1) + E(i,j)\}$$

for $j - i < W$, where W is the maximum allowed haplotype block length.

When the DP algorithm was applied to the ST1, ST2 and ST3 datasets with the maximum allowed block size being 12, we obtained an average of 6 markers per block with the smallest block being a single marker and the largest equal to W. On average, we had 22 distinct haplotypes per block with their number ranging from 1 to 30.

Our algorithm is based on genotypes as opposed to haplotypes that were used in [34]. In the method proposed in [34], each genotype segment was first phased separately and the entropy

Average Transmission Error Rate			
	ST1	ST2	ST3
TDS	0.0039	0.0065	0.0320
Equal TDS	0.0113	0.0085	0.0360

Table 2.9: Average transmission Error Rate For Equal Block Partitioning TDS(Equal TDS).

Incorrect Trio			
	ST1	ST2	ST3
TDS	0.95	1.6	5.4
Equal TDS	1.6	1.7	5.6

Table 2.10: Average number of Incorrect Trios per dataset For Equal Block Partitioning TDS(Equal TDS).

of each block was calculated from the number of occurrences of each unique haplotype in that segment. The same DP algorithm was then applied to the segments and the minimum total block entropy partition was calculated. In order to avoid this time consuming procedure (it can result in computational times even bigger than PHASE) we create the blocks based on the genotypes that can be done instantly. Clearly the bigger the dataset the more accurate our genotype approximation results will be. However, even for small datasets this approach has been shown to improve our results compared to the standard equal block partitioning as shown in Tables 2.9 and 2.10.

2.4.8 Partition-Ligation

In the partition phase the dataset is divided into small segments of consecutive loci using the haplotype block partitioning method described. Once the blocks are phased, they are ligated together using the following method (an extension of the original method described in [28]).

The result of phasing for each block is a set of haplotype solutions, paired with their associated weights. Two neighbouring blocks are ligated by creating merged solutions from all combinaitons of the blocks solutions, each associated with the product of the individual weights, called the

ligation weight. The TDS algorithm is then repeated in the same manner as it was for the individual blocks. However, the weights of the solutions are scaled by the associated ligation weight for that solution. In this way, no information content is lost in the process of ligating.

Furthermore, the order in which the individual blocks are ligated is not predetermined. We first ligate the blocks that would produce in each step the minimum entropy ligation. This procedure allows us to ligate first the most homogenous blocks so that we have more certainty in the solutions that we produce while moving in the ligation procedure.

2.4.9 Summary of the proposed algorithm

In the partition phase the dataset is divided into small segments of consecutive loci using the haplotype block parititioning.

Routine 1

- Enumerate the set of all possible haplotype vectors, Z, based on the given dataset G.
- Initialization: Find all possible haplotype assignments for each trio and rearrange the trios in ascending order according to the number of distinct haplotype solutions each one of them has. Use the first n trios to enumerate all the possible streams, where n is the largest number such that the total number of streams enumerated from the n subjects does not exceed K, and compute their weights
- Update: For $i = n+1, n+2, \ldots$
  - Find the K^{ext} possible haplotypes compatible with the genotype of the i^{th} trio.
  - For $k = 1, 2, \ldots, K^{ext}$

    * Enumerate all possible stream extensions $H_i^{(k,j)} = [H_{i-1}^{(k)}, h_j]$ with $h_j = \{h_{j,1}, h_{j,2}, h_{j,3}, h_{j,4}\}$
    * $\forall j$ compute the weights $w_i^{(k,j)}$ according to 2.5
  - Select and preserve K distinct sample streams $\{H_i^{(k)}, k = 1, \ldots, K\}$ with the highest importance weights $\{w_i^{(k)}, k = 1, \ldots, K\}$ from the set $\{H_i^{(k,j)}, w_i^{(k,j)}, k = 1, \ldots, K, j = 1, \ldots, K^{ext}\}$
  - $\forall k$, update the sufficient statistics $T_i^{(k)} = T_i(T_{i-1}, h_i^{(k)}, g_i)$

TDS Algorithm

- Partition the genotype dataset G into S subsets using the procedure described in the Haplotype Block partitioning subsection.
- For $s = 1, \ldots, S$ apply Routine 1 so that all segments are phased and for each one keep al the solutions contained in the top K streams
- Until all blocks are ligated
  - Find the blocks that if ligated would produce the minimum entropy
  - Ligate the blocks, following the procedure described in the Partition-Ligation section

Chapter 3

Haplotype Inference in General Nuclear families

3.1 Introduction

Computational methods for haplotype inference fall into two large categories. The first category consists of methods that use population data [21] and is used in association studies. Available algorithms include PHASE [23], BEAGLE [30], TDS [35], HAPLOTYPER [28], HAP2 [26, 36]. The second category consists of methods using pedigree data and is mainly used in linkage studies. Several programms for haplotyping related individuals exist, many of which are based on the Lander-Green algorithm [37] and employ different optimization techniques such as MERLIN [38], GENEHUNTER [39], ALLEGRO [40,41], HAPI [42] or are based on a set of genetic rules such as MRH [43], ZAPLO [44], HAPLORE [45] and on Bayesian network representation of the pedigree such as SUPERLINK [46]. Comparative reviews for these two categories have been done [3, 21]. Population based methods attempt to take advantage of features of the population samples to

perform the haplotype inference. On the other hand, methods that use familial data [3] attempt to capture the Identical By Descent (IBD) information inherent in the pedigree structure to perform the phasing.

A scenario that falls in between the population and pedigree based phasing occurs when the inference is performed in trio families, as discussed in the previous chapter, and many population-based methods have been extended to handle that case. This trio inference setup can be seen as a special instance of the more general and less studied problem of haplotype inference in datasets including nuclear families with an arbitrary number of children. Such datasets that include families with more than one child are encountered in sib pair studies, but it is also possible that, in consortiums for complex disorders, families have more than one child genotyped and possibly other family members as well. In these multiple children families, pedigree methods fail to take advantage of the underlying population information to resolve any ambiguities in the parents not resolved by information from the remaining family members. On the other hand, when using any of the well-established trio methods, a family with n children should either be broken into n trios and each one of them examined separately, or we should use only one child from each family to determine the haplotype of the parents. An obvious drawback of the first case is that each of the trios, that have resulted from the original family, may give different solutions for the haplotypes of the parents, which could cause obvious problems, e.g. in an affected sib-pair study. Irrespectively of its use, the fact that different chromosomes are implied for the same parents depending on which child we choose, by itself poses concerns for the accuracy of the downstream analysis. In both cases mentioned above, population trio methods fail to take into consideration simultaneously all the constraints for the parental haplotypes leading to significant loss of valuable information. Furthermore, trios do not provide the ability to detect recombination events.

In a similar study [26], the ability to increase the accuracy in haplotype inference when including a child or more in families, as opposed to phasing the relatives as unrelated individuals, has been demonstrated. However, this study was focusing on small chromosomal segmnets and there were instances that this methodology could not handle that are common with current genome wide association data.

In this chapter we first expand the applicability of the TDS algorithm presented in the previous chapter so that it can handle general nuclear families with any number of children. We examine and validate our approach on scenarios where fewer families are available as opposed to the number of families examined in our previous chapter (and in [21]), since for that number of families the accuracy achieved is very good. We demonstrate the capacity of our algorithm on a real diabetes dataset. Our method exploits simultaneously all the constraints implied by Mendelian inheritance on top of the structure of our previously presented TDS trio framework. We further aim to demonstrate and quantify the gain in the accuracy offered by applying our proposed methodology as opposed to any of the conventional trio configurations we described above.

3.2 Methods

The structure of this section is as follows: For the convenience of the reader we first give a very brief description of the TDS procedure presented in the previous chapter and we put in place the proposed extensions for the case of general nuclear families.

We then describe how to obtain locally for each family the haplotype assignments in the parents that can produce the children with the minimum number of recombination events and illustrate the advantages offered when taking into consideration all the children simultaneously in a family as

opposed to breaking the family into separate trios. We further present the modified version of the partition-ligation procedure adjusted for nuclear families with multiple children so that we are able to handle large datasets and we give a summary of the proposed algorithm. Finally, we present the datasets and figures of merit used.

3.2.1 TDS Algorithm in families revisited

Our tree based deterministic structure examines all possible solutions in each family. In this chapter we allow a family to have multiple children and we denote as $g_t = \{g_{t,f}, g_{t,m}, g_{t,c1}, \ldots, , g_{t,cn}\}$ the genotypes of the t family where $g_{t,f}, g_{t,m}, g_{t,ci}$ stand for the genotypes of the father, mother and child i respectively. Let us assume that family t has N possible solutions and we denote the haplotypes of the parents in the j^{th} solution by $h_t^j = \{h_{t,1}^j, h_{t,2}^j, h_{t,3}^j, h_{t,4}^j\}$, so that similar to the previous chapter $\{h_{t,1}^j, h_{t,2}^j\}$ are the two haplotypes of the first parent and $\{h_{t,3}^j, h_{t,4}^j\}$ are the two haplotypes of the second parent. This set of possible haplotype solutions is derived as will be described in the next subsection "Minimum recombinant orientation in families". Using the same notation as in the previous chapter suppose further that $G_t = \{g_1, ..., g_t\}$ is a set of genotypes of families up to and including family t, where in each family $h_t = \{h_{t,1}, h_{t,2}, h_{t,3}, h_{t,4}\}$ are the haplotypes of the parents and similarly define $H_t = \{h_1, ..., h_t\}$. Families in our algorithm are processed sequentially one at a time. Assume that by the time we have processed genotype g_{t-1} we have a set of possible solutions and their associated weights $\{H_{t-1}^{(k)} | w_{t-1}^{(k)}, k = 1, \ldots, K\}$. Each one of these solution streams has one and only one solution for each family encountered. As explained in the previous chapter, intuitively, the weight that is associated to each solution stream can be interpreted as the probability of this solution stream to contain the actual haplotype solutions

for families up to the t^{th} family.

Suppose now that family t has N possible solutions. We append each of these solutions to the end of each of the previous streams getting a new set of $K*N$ solution streams, $[H_{t-1}^{(k)}, h_t^{(j)}]j = 1,\ldots,N$ for up to family t. The weight for each of the streams is a product of the weight this stream had prior to attaching one of the possible haplotype orientations for family t by a term representing how likely it is to see solution h_t^j for family t given that the solutions for the previous families are $H_{t-1}^{(k)}$ and is calculated as

$$w_t^{(k,j)} \propto w_{t-1}^{(k)} \frac{B(\rho^{(k)}(t-1)+r)}{B(\rho^{(k)}(t-1))} \tag{3.1}$$

in which $r = [\sum_{i=1}^{4} I(z_1 - h_{t,i}^j), \ldots, \sum_{i=1}^{4} I(z_M - h_{t,i}^j)]$, $\rho^{(k)}(t-1)$ is the parameter vector of the assumed Dirichlet prior which represents how many times we have encountered each haplotype in stream k in the solutions up to family $t-1$, M is the total number of encountered haplotypes and $B(\rho^{(k)}(t-1)) = \frac{\prod_{i=1}^{M}\Gamma(\rho_i^{(k)}(t-1))}{\Gamma(\sum_{i=1}^{M}\rho_i^{(k)}(t-1))}$.

According to the weights of each stream, we keep only a subset K of the $K*N$ streams that have the highest weights.

We note here that, as evidenced from our previous description of the TDS scheme, the solution selection and the weight assignment when the t^{th} family is processed depends on the previously examined families. To make our assignment more accurate we would be benefited, if the uncertainty for the haplotype assigment in the previously encountered families is minimum. Therefore, in our algorithm the order in which we process the families on each block is not predefined. We initially find all possible haplotype assignments for every family. We then put the families in an ascending order according to the number of distinct solutions each of them has and process them accordingly.

The whole procedure is deterministic. The reason that we choose to process the families in such a way is because the families with fewer solutions imply less uncertainty. Moreover, it is usually the case that many families have only one possible haplotype assignment. In these families there is no uncertainty and it makes sense that these families should be processed first, since they represent haplotype segments that we are sure exist in the population.

3.2.2 Minimum Recombinant Orientation in families

In our algorithm we partition the dataset into small blocks of consecutive loci that exhibit limited haplotype diversity, as will be described in the partition-ligation section. This subsection describes our procedure for identifying minimum recombinant orientations for a family in a specific block. Suppose that $g_t = \{g_{t,f}, g_{t,m}, g_{t,c1}, \ldots, , g_{t,cn}\}$ are the genotypes of the t family as defined in the previous subsection where each family member is genotyped in L SNPs. In the trio case, the only constraints in the set of compatible haplotypes in the family were stemming from laws of Mendelian inheritance. So the set of compatible solutions would consist of all haplotypes that could produce the genotypes of the parents, and had a combination of haplotypes one from each parent that could produce the child genotype. When a family has more than one child, a haplotype solution for a specific trio is still a valid haplotype assignment to the parents. However, each different such assignment to the parental haplotypes demands a different number of recombination events to produce the haplotypes of the remaining children. In our approach, we have considered from each family only the set of possible parent haplotype assignments that produce haplotypes that have locally minimal numbers of recombinations in the children.

As we mentioned above and as wil be described in the partition-ligation subsection, in our al-

gorithm we partition the dataset into small blocks of consecutive loci that exhibit limited haplotype diversity. It is clear that within each block, for the vast majority of the cases there would be haplotype assignments to the parents that can produce the children without assuming any recombination event and thus they would consist of the minimum recombinant set of solutions for the family. So as an optimization step, in each family with more than one child, we first consider all the haplotype assignments to the parents stemming from the trio that includes the child with the minimum triple heterozygote (ambiguous) sites. All solutions of that trio that can produce the remaining children consist of valid non-recombinant haplotype assignments to the parents and thus valid minimum recombinant orientations. This optimization step has allowed us to find, if they exist, valid non-recombinant solutions for that specific family. In the case that no solution could be obtained as described above, a straightforward way to obtain a set of solutions implying the minimum number of recombination events is as follows: First we break each family with n children into n trio families and phase each one of them separately. Let us assume that trio j has n_j possible solutions and we denote the haplotypes of the parents in each solution by $\{\tilde{h^j_{i,1}}, \tilde{h^j_{i,2}}, \tilde{h^j_{i,3}}, \tilde{h^j_{i,4}}\}$ so that $\{\tilde{h^j_{i,1}}, \tilde{h^j_{i,2}}\}$ are the two haplotypes of the first parent and $\{\tilde{h^j_{i,3}}, \tilde{h^j_{i,4}}\}$ are the two haplotypes of the second parent, and that the haplotypes transmitted to the child from each parent are $\tilde{h^j_{i,1}}$ and $\tilde{h^j_{i,3}}$, respectively. Therefore, the number of possible haplotype assignments to the parents are $P = \prod_{i=1}^{n} n_i$.

For each of these combinations of haplotype assignments in the children, we can reconstruct the possible haplotype orientations of the parents that can produce them with the minimum number of recombination events. To illustrate how we find this minimum recombinant orientation in the parents, we observe that since we have for each child the haplotype inherited from each parent we can perform the procedure in each parent independently. Suppose now that we consider one of these P such possible assignments in which the haplotypes transmitted to the children from

the first parent are $\tilde{h_2^j}$ $j = 1, \ldots, n$ according to our previous notation. To create the haplotype configurations of the parents that would produce the children with the minimum number of recombination events, we start building the haplotypes of the parents from the beginning of the block SNP by SNP. Homozygous SNPs will not impose a difference in the recombination events since both haplotypes will have the same allele for these SNPs. At each heterozygous marker we choose the chromosome that each allele is going to be assigned according to which assignment imposes the minimum recombination events at that marker and maintaining both assignments if they produce the same number of events. To find that, we just have to count the number of recombination events imposed on that exact SNP under both assignmetns. If both assignments produce the same number of recombination events, we have to keep both of them and perform the procedure for the remaining SNPs in each one of them separately. Since all P solution combinations are processed sequentially, if at any point in the current combination the number of recombination events exceeds the minimum number we have already encountered (from a previous combination) then the procedure is automatically terminated for that combination and we move to the next one. At the end of this process, family t will have a corresponding set of parental haplotype assignments.

An illustrated example of our procedure on a family with three children genotyped on three SNPs is given in Figure 3.1. In the first step (A), the three-children family is broken into three separate trio families and each one of them is phased separately. Since none of the solutions of the second trio can produce the remaining two children we cannot obtain a non-recombinant solution and we have a recombination event. In the second step (B), we consider all possible solution combinations (1a-2-3a, 1a-2-3b,1b-2-3a,1b-2-3b) and demonstrate the procedure for combination 1b-2-3b. We consider as known what is transmitted from each parent and we create the haplotypes

of the parents SNP by SNP. In the first parent the first SNP is heterozygous and in all children the first haplotype (1) is transmitted. The second SNP is homozygous. For the third SNP if we assign 1 on the first haplotype we have to assume two recombination events to get the haplotypes of the children and if we assign it on the second one recombination event. We repeat the procedure in the second parent and we choose the solutions implying the minimum number in each one of them. The total number of recombination events is the sum of the events in both parents. In the third step (C), after repeating the procedure in all solution combinations, we keep the ones with the minimum total number. In this example we have two solutions with one recombination event that both produce the same parental configuration (which is 1a).

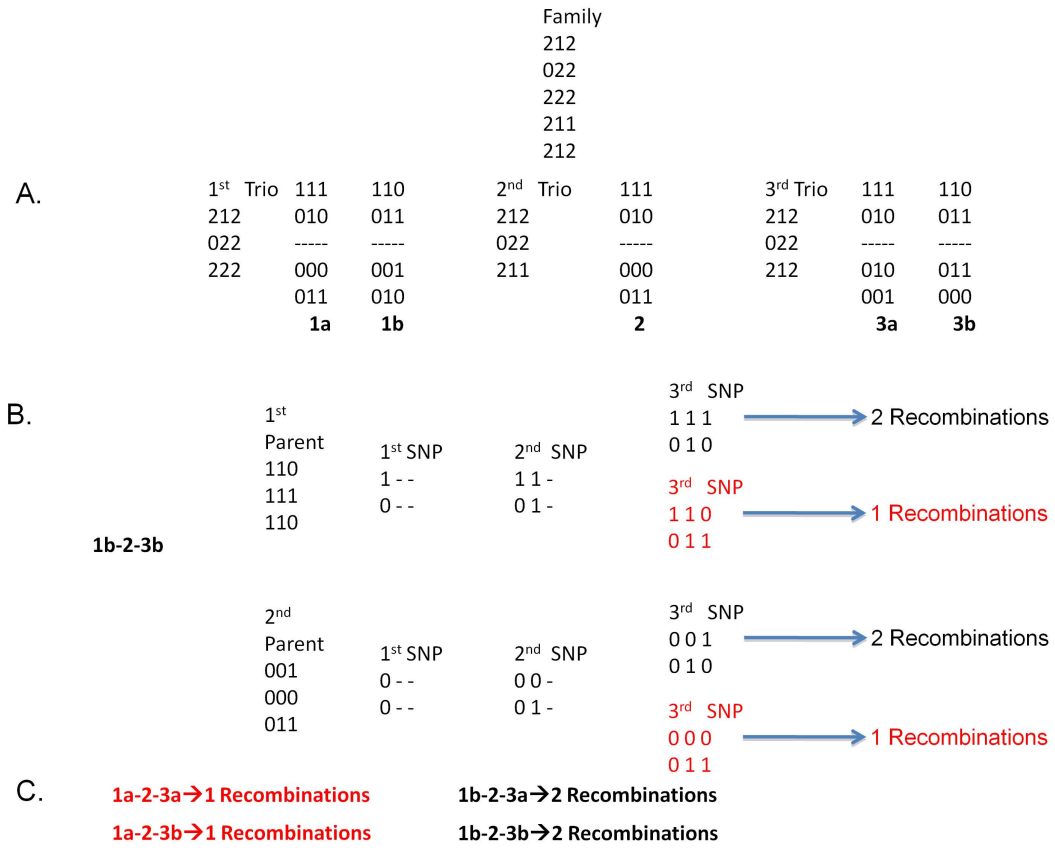

Figure 3.1: Illustration of the procedure for identifying minimum recombinant orientations in a family

3.2.3 TDS trios and multiple children comparison

In this section we give an illustrative comparison between the trio and the multiple children methodologies in a given family, highlighting the main advantages when taking into consideration all the children simultaneously. Suppose, that we denote the major allele in a particular SNP locus in a haplotype as ”0” and the minor allele as ”1”. Similarly, in a genotype we denote by ”0” that the individual is homozygous to the major allele at that SNP and with ”1” that the individual is homozygous to the minor allele. We denote by ”2” the heterozygous case.

In Figure 3.2, we emphasize the advantages offered when considering all children in a family simultaneously. We focus on the expansion of one stream when a family with three children is encountered and contrast the expansion performed when the family is broken into three separate trios as opposed to a single three-child family.

Intuitively, we can discern which solution for the first child is the correct one when considering the second child. However, we cannot take advantage of that fact when the family is broken into trios that are phased irrespectively of each other. The third child is a recombinant child. None of the solutions produced when it is phased irrespectively of its siblings are correct (Figure 3.2 solutions 3a, 3b). This means that none of the resulting streams would correctly phase all three children, while only two streams phase correctly the first two children (H|1a| 2|3a, H|1a|2|3b). Furthermore, the parents are assigned different haplotypes in each phasing.

When considering all trios together, we can end up with the correct haplotype solution based on the minimum recombination criterion. Even though our algorithm searches through all possible parental haplotype orientations to find the one that would produce the children with the minimum number of recombination events, we can see that if, e.g., the actual solution for the parents was

1b, then 3a or 3b would need two recombination events to obtain the haplotypes of all children. Moreover, considering all the children simultaneously, we are also able to identify the third child as a recombinant child.

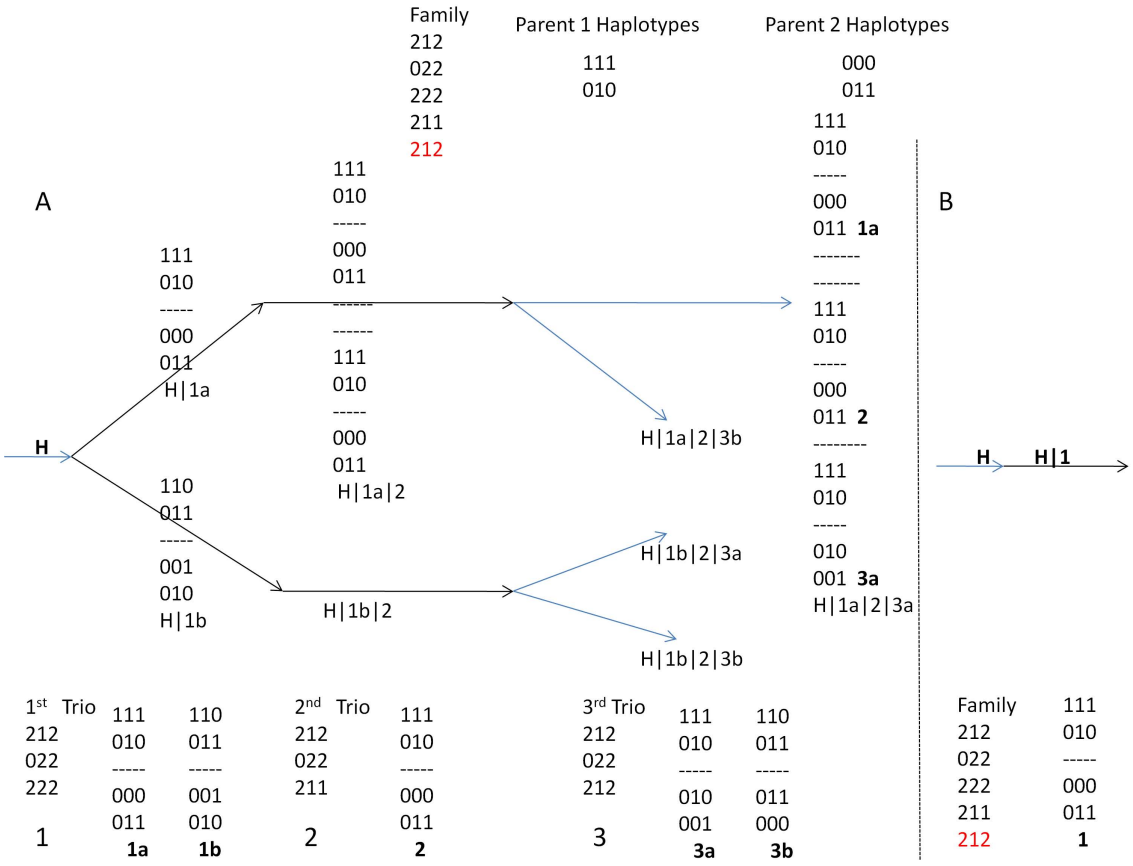

Figure 3.2: A family with three children is phased using TDS (A) broken into three separate trios and (B) considered as a single family. The actual parental haplotypes are given next to the family. For each trio the possible solutions are given next to the genotypes of the trio at the bottom of (A). In each of the possible solutions for each trio the first two haplotypes are the transmitted and untransmitted haplotype from the first parent and similarly the remaining two for the second parent. (A) The family is broken into three separate trios that are phased sequentially. All solution combinations are displayed. In our algorithm each solution stream ($H|1a|2|3a$, $H|1a|2|3b$, $H|1b|2|3a$, $H|1b|2|3b$) will be assigned a weight and the one with the highest will be selected. (B) The family is phased as a multi-children family and only one solution is retained based on the minimum recombination criterion.

3.2.4 Partition Ligation

In the partition phase, the dataset is divided into small segments of consecutive loci using the haplotype block partitioning method described in the previous chapter and each of the individual blocks is phased separately. To ligate the individual blocks we have adjusted the corresponding ligation method presented in the previous chapter as will be described below.

After phasing each block we have a set of haplotype solutions for each family, paired with their associated weights. Two neighbouring blocks are ligated by creating merged solutions from all combinations of the block solutions, each associated with the product of the individual weights, called the ligation weight. This full set of possible solutions is then filtered down to a set of solutions that implies a minimum number of recombination events as follows:

In each family when creating a merged solution from a solution combination, one from each adjacent block, we have to decide which haplotypes for each parent will be assigned on the same chromosome.

As opposed to the trio case where the transmitted haplotype from each parent is known and there is no ambiguity on how the haplotypes are going to be aligned on the chromosomes, in the multiple children families we have to do an extra step in ligating the haplotypes in two neighbouring blocks in a family. Depending on which haplotypes from each block are going to be assigned on the same chromosome for each parent, a different number of recombination events will occur when creating the genotypes of the children. Again we use the same principle, that is the minimum number of recombination events, to decide which haplotypes, one from each block are going to be assigned on the same chromosome. From the set of all possible solution combinations in each family in the end we consider only the ones that imply the minimum number of recombination events

and the weights associated with them. The TDS algorithm is then repeated in the same manner as it was for the individual blocks with the weights of the solutions scaled by the associated ligation weight for that solution.

Furthermore, the order in which the individual blocks are ligated is not predetermined. We first ligate the blocks that would produce in each step the minimum entropy ligation. This procedure allows us to ligate first the most homogenous blocks so that we have more certainty in the solutions that we produce while moving in the ligation procedure.

3.2.5 Simulated Data

We used two different kinds of simulated datasets. For consistency, the first three datasets were the ones provided in [21]. The haplotypes were simulated using a coalescent model COSI [31] that incorporates variation in recombination rates and demographic events and the parameters of the model were chosen to match aspects of data from a sample of white Americans. The first two datasets were simulated with constant and variable recombination rates respectively and the third dataset had a variable recombination rate and demography parameters consistent with that of white Americans. Each of the three datasets provided consisted of 20 sets of 30 trios spanning 1Mb of sequence with a density of one SNP per 5 Kb.

We have also used the COSI software to create our own realistic simulated data sets, each consisting of 2,000 haplotypes with 1Mb of marker data using the ”best-fit” parameters obtained from fitting a coalescent model to the real data. Samples were taken from a European population and each simulated data set has a recombination rate sampled from a distribution matching the deCODE map [32], with recombination clustered into hotspots. For each simulated data set, we

initially selected only those markers with minor allele frequency greater than 0.05. We then applied a greedy tag-SNP selection algorithm [47] on these markers to retain a maximally informative set of SNPs so that each marker in the initial dataset was either genotyped or had r^2 above a specific threshold with at least one of the selected genotyped markers. We considered these threshold values to be 0.7 (low density set) and 0.9 (high density set). The median number of markers for the low density set was 236 markers with a range of 30-652 markers, whereas in the high density set the range in the number of markers was 61-922 with a median of 389 markers.

To consider a larger number of markers we have used the COSI software and the same configuration of parameters described above to create datasets consisting of 2,000 haplotypes with 20Mb of marker data. We have then applied the same procedure as the one applied to the 1Mb datasets and considered the r^2 value to be 0.9. Finally, from each dataset we considered exactly 8000 markers.

In the three datasets provided, the families had been already created. In each family we considered the haplotypes of the parents and created the children by randomly choosing a transmitted haplotype from each parent and considered the same recombination rate as that of [26] a rate of 30 recombination events per $3x10^9 bp$. In the datasets that we created, each parent chose a haplotype from the pool of haplotypes and the procedure for creating the children was exactly the same as before.

3.2.6 Real Data

We have applied our method to data from the Type 1 Diabetes Genetics Consortium (T1DGC), which consists of 2,300 families with 9,749 individuals from nine cohorts, namely Asia Pacific

(AP; 191), British Diabetic Association (BDA; 418), Danish (DAN; 147), European (EUR; 475), Human Biological Data Interchange (HBDI; 431), Joslin Diabetes Center (JOS; 112), North American (NA; 334), Sardinian (SAR; 78) and United Kingdom (UK; 114). To have a coherent pool of families we chose the largest subgroup in terms of individuals and families based on the primary, secondary and tertiary ethic groups consisting of 1,675 British individuals from 392 families. From the total number of families, 19 families had only one child, 272 families had two children, 83 families had three children and the remaining 18 families had more than 3 children.

The fine mapping data include 2,957 SNPs in the MHC genotyped on two oligonucleotide pool assays (OPA) using the Illumina Golden Gate platform at the Wellcome Trust Sanger Institute. Quality control on the SNP genotypes were performed using PLINK [48]. First, any Mendelian errors were set as missing. Then SNPs with call rate < 0.95 and $MAF < 0.01$ were discarded. The remaining 2,338 SNPs were tested for HWE and those with $p < 0.05/2,338 \sim 2.14x10^{-5}$ were removed, resulting in a total of 2,259 SNPs. Information on access to data and samples is available at (https://www.t1dgc.org/home.cfm), and details of the dataset can be found in [49].

We have also applied our method on artificially created multiple children families from the HapMap data. Similarly to the simulated data we have created datasets with varying number of markers. First, we have randomly selected 80 non-overlapping 1Mb regions from Chromosome 1 from the CEU HapMap population (HapMap 3 release 2- Phasing data). We have then considered 25 non-overlapping 40Mb regions from chromosomes 1,2,3,4,5 and 8 from the same population. In each region, we initially selected only those markers with minor allele frequency greater than 0.05 and then randomly selected markers to obtain a density of approximately one marker per 5kb. For the 40Mb region datasets we fixed the number of markers to 8000 so that it is in accordance with the number of markers in the simulated datasets. Finally, we have considered the full set of

markers from each chromosome, without any filtering. We have again used the CEU population and for each chromosome we have created datasets of 10 families each. There were no overlapping families among the datasets. The number of markers for each chromosome is shown in Table 3.1. For each family, we created extra children using the same procedure we described for the artificial data.

3.2.7 Measurement of phasing accuracy

We have used two measures for the accuracy of phasing. The switch error rate [21, 36] is the percentage of switches among all possible switches in haplotype orientation used to recover the correct phase in an individual. In datasets with missing SNPs the imputation error rate [30] is defined as the proportion of mistakenly inferred alleles among all missing alleles.

3.3 Results

3.3.1 Switch error rate

In the absence of real phased data, to evaluate the switch error rate all comparisons between the different scenarios and algorithms were performed on simulated data. We created families with three children and we considered all possible phasing configurations for obtaining the haplotypes of the parents, which are the independent samples from the population. We have used our previous trio algorithm (TDS [35]) and BEAGLE [30] and we have phased the data with the two options available, when only considering trio algorithms: either taking only one child from each family or breaking down the family with three children to three independent trios. A possible problem with

the second option, where the family is broken into three separate trios, even though it is expected to give more accurate phasing results as opposed to considering only one child from each family, is that many times for the same parents different haplotype assignments are implied from the different trios of the same family. Using our modified extension, we have considered the scenarios of taking two out of the three children in each family simultaneously and finally all three children together. In all cases the final error rate was an average of the switch error rate of the parents in each family. The results are shown in Figure 3.3. As expected, the worst performing methods, when considering the average of the switch error rates across all families, are TDS and BEAGLE when we use only one child. The performance of both algorithms increases when all three separate trios from each family are used for the phasing. The best performance in all datasets is achieved when all three children are used simultaneously for the inference.

Furthermore, we have evaluated our method on whole chromosomal segments. For each chromosome, we have considered the full set of markers and we have created four datasets each one consisting of 10 families with three children as described in the ”Methods” section. We have phased these datasets using two alternative scenarios. First we have used our modified extension so that we simultaneously consider all children in a family. We have then broken down each family with three children to three independent trios and performed the phasing using the BEAGLE trio algorithm. The results (along with the number of SNPs on each chromosome) are presented on Table 3.1 showing, consistently with the simulated data, that we can achieve better performance when all children are used simultaneously for the phasing.

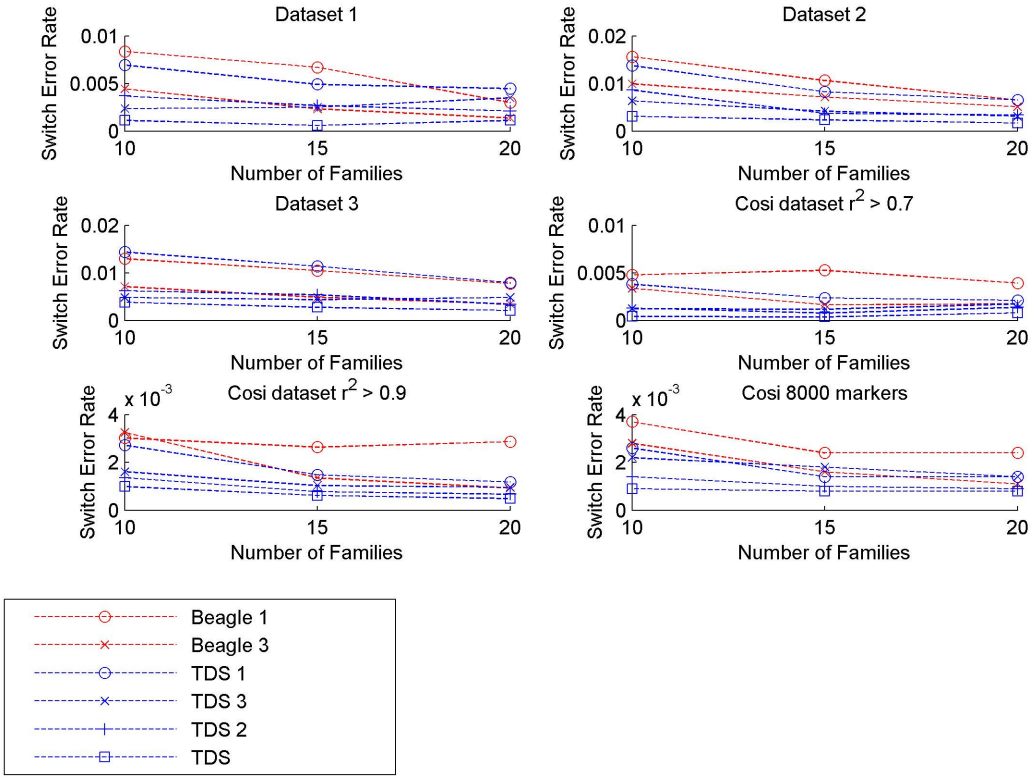

Figure 3.3: Switch error rate. Estimating error rates by considering only one child from each family with BEAGLE and TDS (BEAGLE 1, TDS 1), all three children as three separate nuclear families (BEAGLE 3 and TDS 3) and two and three children simultaneously in a multi-children family (TDS 2 and TDS).

Chr	Number of Sites	Switch error rate(%)TDS	Switch error rate(%)Beagle
1	116,415	0.08	0.19
2	116,430	0.11	0.26
3	96,537	0.16	0.37
4	85,772	0.09	0.22
5	87,919	0.12	0.27
6	91,357	0.14	0.35
7	75,320	0.16	0.38
8	75,272	0.19	0.46
9	63,612	0.11	0.31
10	73,832	0.14	0.33
11	70,973	0.16	0.35
12	68,525	0.18	0.39
13	51,915	0.17	0.39
14	45,474	0.1	0.24
15	42,353	0.23	0.47
16	44,648	0.14	0.38
17	38,401	0.11	0.26
18	40,824	0.15	0.32
19	26,238	0.27	0.5
20	36,258	0.17	0.36
21	19,306	0.24	0.53
22	20,085	0.08	0.15
Total	1,387,466		

Table 3.1: Switch error rate. Estimating switch error rates with BEAGLE and TDS for datasets including the full set of markers in each chromosome.

3.3.2 Imputation error rate

For consistency, we have first evaluated the performance of our algorithm on the same simulated datasets we have used for the evaluation of the switch error rate. To show the applicability of our methodology to real data, we have also evaluated the performance of our algorithm on three children families created from HapMap data. We have assigned randomly a realistic percentage of 2% of SNPs to missing values. We have imputed the values of the missing SNPs in the parents using two different scenarios. First we have used our augmented framework considering all three children simultaneously. We have then broken each family to three separate trios and performed the imputation using BEAGLE, which from our experience is one of the fastest and most accurate methods for phasing and imputation. The results are shown in Figure 3.4, demonstrating that our algorithm, using information from all children simultaneously, performs better for all family sizes with special emphasis on the cases where the number of families is small. We have also compared the previous two scenarios in the real T1DGC dataset. In this dataset 1% of the SNPs is missing. We have masked another 1% of the SNPs to missing values. We have then randomly considered sets of 15 families phased with our methodology and broken down to trios and phased with BEAGLE. The number of children in the families varied as described in the Methods section and subsets of families were created randomly. The same imputation in the parental genotypes is performed and the imputation error rate when using all children simultaneously with TDS was 0.0123 compared with 0.0156 when broken to trios and phased with BEAGLE, demonstrating the improved accuracy of our methodology in uncovering the missing SNPs.

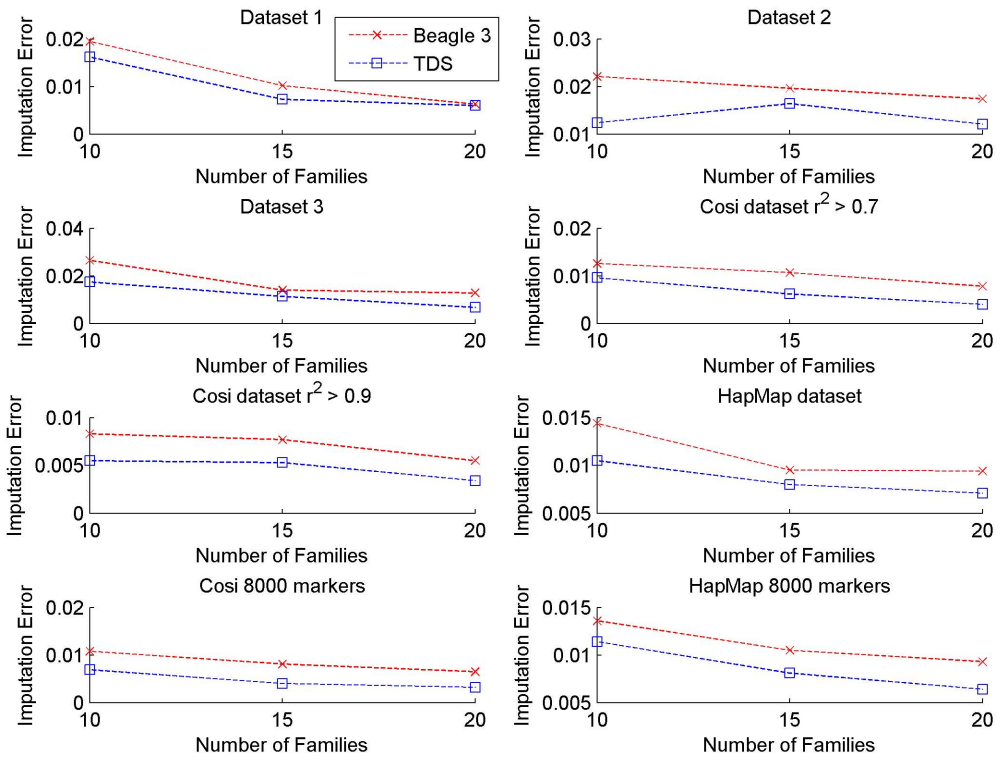

Figure 3.4: Imputation error rate. Imputation error rate for three children families with TDS and BEAGLE as three separate nuclear families (BEAGLE 3) after setting 2% of the SNPs to missing values.

3.3.3 Inference with pedigree based methods

In the previous two subsections we have compared our methodology against trio inference scenarios that provide deterministic assignments for all markers in a dataset. For consistency we evaluate a well known pedigree based method MERLIN, on the T1DGC dataset with the same 1 of masked alleles as in our previous subsection. For our methodology we consider the same partitioning of the T1DGC dataset as described in the previous subsection. For Merlin, similar to all pedigree based methods each nuclear family is phased in isolation and we have obtained haplotypes corresponding to the most likely pattern of gene flow (–best command line option). We have then examined the haplotype inference and missing SNP imputation performance.

We note here that as opposed to the previous trio inference scenarios where the values of all missing SNPs are inferred and all heterozygous SNPs are assigned to their respective chromosomes, when using MERLIN a percentage of missing alleles is not recovered and heterozygous SNPs may have ambiguous phasing. MERLIN like all pedigree based algorithms examines each single nuclear family in isolation ignoring the rest of the families in the dataset. As such there can be ambiguous or not resolved instances in the haplotype inference and imputation scenarios. In particular when assigning alleles in the chromosomes these instances could include heterozygote loci in a parent that both alternative assignments in its respective chromosomes are equally probable (or indistinguishable) or loci where both assignments result in the same number of recombination events. In the presence of missing SNPs these instances include also missing alleles that cannot be imputed from the values of the remaining family members.

For the T1DGC dataset, MERLIN, as argued above, was not able to infer approximately 36% of the masked alleles(12,141 alleles). For the remaining 64% of the alleles their majority were in-

		TDS	
		Correct	Incorrect
MERLIN	Correct	21.345(98.6%)	7 (0.03%)
	Incorrect	288(1.33%)	9(0.04%)

Table 3.2: Missing allele inference with MERLIN and TDS. Correct and incorrect calls with MERLIN and TDS on 64% of the alleles (21,649 alleles), which MERLIN was able to impute.

ferred correctly by both methods even though a small percentage was mistaken by MERLIN(Table 3.2). We have noticed that the vast majority of inferred alleles mistaken only by MERLIN correspond to heterozygous inferred loci for which haplotype assignment was ambiguous . Excluding all such SNPs and considering them as missing, essentially erases almost all mistakes produced only by MERLIN, as shown in Table 3.3, resulting however in a higher percentage of not recovered alleles. We further note that for the actual missing SNPs in the dataset the percentage of alleles not recovered by MERLIN was 50%.

Furthermore, there was a percentage of heterozygous loci for which MERLIN produced ambiguous phasing as it could not determine their haplotype assignment. Clearly loci that are heterozygous for all family members belong to that category. The mean number of ambiguous sites in a nuclear family according to the number of children in the family is shown in Table 3.4. There were on average 206 SNPs in each trio family (on a total of 2259 SNPs)for which phasing was ambiguous.The number of ambiguous sites dropped to 96 sites per family when the family had two children and was further reduced to 41 for three children families and 12 when the family had more than three children. Intuitively the more children in a family that are genotyped, the more probable it is that a child exists that can resolve the ambiguity in a given locus and thus fewer ambiguous sites exist. For families with more than three children, as demonstrated in Table 3.4, we expect MERLIN to produce ambiguous phasing in only a small number of markers.

		TDS	
		Correct	Incorrect
MERLIN	Correct	20.830(99.92%)	5 (0.02%)
	Incorrect	2(0.01%)	9(0.04%)

Table 3.3: Missing allele inference with MERLIN and TDS. Correct and incorrect calls with MERLIN and TDS on only those alleles MERLIN was able to impute, excluding heterozygous inferred missing SNPs for which MERLIN produced ambiguous phasing.

Number of children	Number of families	Average number of ambiguous sites per family
1	19	206
2	272	96
3	83	41
More than 3	18	12

Table 3.4: Average number of ambiguous sizes for families when phasing is performed with MERLIN

3.4 Discussion

To infer the haplotypes of all individuals in a set of nuclear families all pedigree-based algorithms process each single nuclear family independently (or pedigree in the general case) ignoring the remaining families in the dataset. Therefore, there exist instances in the hapltoype inference problem in nuclear families, as demonstrated in the ”Results” section, where these methods fail to provide a solution and we therefore need to consider the underlying population information to obtain a definite solution. On the other hand population-based methods and in particular trio-based methods fail to simultaneously account for all available familial information in a nuclear family as they can only take into consideration one child at a time. Compared to a pedigree-based algorithm our methodology focuses and resolves all these specidfic instances and we further demostrate that it would be preferable to consider a combination of pedigree-population methods like our method as opposed to resorting to a trio inference scenario.

The objective of our proposed methodology is therefore to enable researchers to derive efficiently haplotype configurations for all individuals in a general study which includes nuclear families some of which may have more than one child with missing genotypes (at a rate of 1 or 2%) and genotype errors. As such it should not be considered as an alternative method for inferring haplotype orientations in a specific isolated nuclear family.

In our algorithm we use the procedure described in the ”Minimum recombinant orientation in families” subsection to identify the minimum recombinant haplotype configurations for a specific family in a specific block that will be used as input in our TDS scheme. Merged solutions from adjacent blocks are created as described in the Partition-ligation subsection which are then subjected to the same TDS scheme. This procedure does not in principle guarantee that the derived final solution will be a minimum recombinant solution across the entire dataset. This is clear if we think that since the TDS procedure in each block is independent of the adjacent blocks, potential minimum recombinant configurations may be discarded within each block during the TDS scheme. Our procedure rather focuses on identifying the minimum recombinant solution sets locally using all the familial constraints and considers this set as the input for our sampling scheme.

An important observation in our experiments is that the relative gain using all familial information simultaneously (compared to a trio inference scenario) increases as the number of families decreases. This means that as the number of families decreases, the familial constraints start playing an important role, which cannot be modelled by population features alone. Thus, even though we generally assume gain in accuracy when taking the familial structure into consideration compared to the scenario where families are broken to independent trios, we expect the gain to diminish, as the number of families grows large.

We would also like to emphasize that both algorithms, TDS and BEAGLE, when presented

with trios from the same family, are able to inherently use this information. The reason is that all trios stemming from the same multi-children family share some common solutions locally and potentially globally, unless there is a recombination event. In TDS all such solutions are going to be assigned greater weight as they are repeated during the course of the algorithm. In BEAGLE, the EM style approach followed will benefit from the repeat of the localized patterns found in the common solutions and will likely identify them.

An important parameter on all algorithms when intended for everyday use is speed. We have shown in the previous chapter that our methodology can attain similar or greater speed compared to BEAGLE. Incorporating the familial constraints has not resulted in important differentiations in the scaling of our methodology.

Chapter 4

Haplotype Inference and Frequency Estimation in Pooled Genotype data

4.1 Introduction

Typically, the first phase of a GWAS includes genotyping across hundreds of individuals and validation of the most significant SNPs. One possible approach to reducing the overall cost of the study is to replace individual genotyping in phase I with allelotyping of pooled genomic DNA [50–55]. Here, equimolar amounts of DNA are mixed into one sample prior to the amplification and sequencing steps. After genotyping, the frequency of an allele in each position is given [54].

Haplotypes are valuable again in this setting as they can improve the power of detecting associations with disease and are also of general interest with the pooled data. To facilitate haplotype-based association analysis it is necessary to estimate haplotype frequencies from pooled DNA data.

A variety of algorithms have been suggested to estimate haplotype frequencies from pooled data. Available methods fall into two large categories. The first category consists of methods that

focus on accurate solutions for small pool sizes (2 or 3 individuals per pool) and considerably large genotype segments. Many well known approaches that focus on small pool sizes use an EM algorithm for maximizing the multinomial likelihood [56–58]. Pirinen et al. [59] extended the gold standard PHASE algorithm [23] to the case of pooled data. They introduced a novel step in the Markov Chain Monte Carlo (MCMC) scheme, during which the haplotypes within each pool were shuffled to simulate individuals on which the original PHASE algorithm could be run to estimate the haplotypes. A method based on perfect phylogeny, HAPLOPOOL, was suggested in [60] and was supplemented with the EM algorithm and linear regression in order to combine haplotype segments. HAPLOPOOL has demonstrated superior performance in terms of accuracy and computational time with respect to the competing EM algorithms. The second category consists of methods that focus on large pools (order of hundred of individuals per pool) and considerably smaller genotype segments. For this scenario, Zhang et al. [61] first proposed a method (PoooL) for estimating haplotype frequencies using a normal approximation for the distribution of pooled allele counts. Imposing a set of linear constraints they transformed the EM algorithm to a constrained maximum entropy problem which they solved using the iterative scaling method. Kuk et al. [62] improved the PoooL methodology, using the ratio of normal densities approximation in the EM, which resulted to the AEM method. Gasbarra et al. [63] introduced a Bayesian haplotype frequency estimation method combining the pooled allele frequency data with prior database knowledge about the set of existing haplotypes in the population. Finally, HIPPO [64] used a multinormal approximation of the likelihood and a reversible-jump Markov chain Monte Carlo (RJMCMC) algorithm to estimate the existing haplotypes in the population and their frequencies. The HIPPO framework is also able to accommodate prior database knowledge for the existing haplotypes in the population and has demonstrated improvements in the performance over the approximate EM - algorithm [64]. In this

chapter we will therefore compare our proposed algorithm with the top performing methods from each category as discussed above, namely HIPPO and HAPLOPOOL.

Naturally, pooling techniques are more prone to errors and offer less possibilities for assessing the quality of the data than individual genotyping. As argued and discussed by Kirkpatrick et al. [60], pooling errors have much greater effect on larger pool sizes as opposed to small pool sizes with respect to the number of incorrect allele calls and the subsequent haplotype estimation. In specific, if σ is the error standard deviation (SD) in the estimates of allele frequencies, $2 * \sigma$ should be less than the difference between allowable frequency estimates, in order for clustering algorithms to be able to correct the error. As more individuals are included in each pool, the difference between allowable allele frequencies decreases, which results in a higher percentage of incorrect calls. For example in pools of two individuals where the difference between allowable frequency calls is 0.25 (0, 0.25, 0.5 ,0.75,1), an accuracy of $\sigma < 0.125$ will ensure a low rate of incorrect calls ($< 1\%$).

In a recent study Kuk et al. [65] examined the efficiency of pooling relative to no pooling using asymptotic statistical theory. They found that under linkage equilibrium (not a typical case!) pooling suffers loss in efficiency when there are more than three independent loci (2^3 haplotypes) and up to four individuals per pool, whereas accuracy decreases with increasing pool size and number of loci. Rare alleles or linkage disequilibrium (or both) decrease the number of haplotypes that appear with non-negligible frequencies and thus pooling could remain efficient for larger haplotype blocks. In general, pooling could still remain more efficient in the case where only a small number of haplotypes can occur with appreciable frequency, as also suggested in Barratt et al. [66], and while pool size is kept considerably small.

In this chapter we introduce a new tree-based deterministic sampling method (TDSPool) for

haplotype frequency estimation from pooled DNA data. Our method specifically focuses on small pool sizes and can handle arbitrarily large block sizes. We present results on real data focusing on dense SNP areas, in which only a small number of haplotypes appear with appreciable frequency, so that our scenarios are within the limits of Kuk et al. [65]. We demonstrate that using TDSPool we can achieve improved performance over existing state-of-the-art methods in datasets with large number of markers.

4.2 Results

In order to compare the accuracy of frequency estimation between the different methods and under the different scenarios examined, we compared the predicted haplotype frequencies from a given method, f, to the gold-standard frequencies, g, observed in the actual population. The measure we used was the χ^2 distance between the two distributions which is simply the result of the χ^2 statistic, where g is the expected distribution, i.e., $\chi^2(f,g) = \sum_{i=1}^{d} (f_i - g_i)^2/g_i$ and d is the number of gold standard haplotypes [60].

4.2.1 Datasets

To examine the performance of our methodology we have considered in our experiments real datasets for which estimates of the haplotype frequencies were already available and which cover a variety of dataset sizes.

We have first simulated using the three loci haplotypes and their associated frequencies from the dataset of Jain et al. [67] as the true distribution (Table 4.1). The haplotypes and their frequencies were estimated using the EM algorithm from a set of 135 individuals genotyped on three SNPs

Haplotype	Frequency
1 0 0	0.082
0 0 1	0.525
1 0 1	0.283
1 1 1	0.106

Table 4.1: Haplotypes and their estimated frequencies for the 3 loci dataset

and the estimates were used as the true haplotype distribution. We have simulated datasets with a variable number of pools T=50, 75, 100 and 150. In each pool each individual was randomly selecting a pair of haplotypes according to the distribution of haplotypes. We have created pools with two different pool sizes, 2 and 3 individuals per pool. For each number of pools and each pool size we have created 100 datasets that were used as the datasets for our simulation.

Next, we considered two more cases with larger number of loci. In the second case which has $L = 10$ loci, we generated data according to the haplotype frequencies of the AGT gene considered in Yang et al. [58]. The haplotypes and their respective frequencies are given in Table 4.2. The procedure for creating datasets and pools was identical to the three loci case.

The third dataset consisted of SNPs from the first 7Mb (742 kb to 7124.8 kb) of the HapMap CEU population (HapMap 3 release 2- Phasing data). This chromosomal region was partitioned based on physical distance into disjoint blocks of 15 kb. The resulting blocks had a varying number of markers ranging from 2-28. For our purposes we have considered only the datasets that had more than 10 SNPs and less than 20 (which was the maximum number of loci so that HAPLOPOOL could produce estimates within a reasonable amount of time) which resulted in selecting a total of 80 blocks. On each block the parental haplotypes and their estimated frequencies were used as the true haplotype distribution. As in the previous cases, in each block two different pool sizes, 2 and 3 individuals per pool, were considered and four different number of pools per dataset.

Haplotype	Frequency
1 1 1 1 0 1 1 0 0 0	0.033
1 1 0 1 0 1 1 1 1 0	0.016
1 1 0 1 0 0 1 0 0 1	0.017
1 0 0 1 0 1 1 0 0 1	0.017
1 1 0 1 0 1 1 0 0 1	0.017
1 1 1 1 0 1 1 1 0 1	0.507
0 1 0 1 1 0 0 1 1 1	0.017
1 1 0 0 0 0 1 1 1 1	0.033
0 1 0 1 0 0 1 1 1 1	0.1
1 1 0 1 0 1 1 1 1 1	0.193
1 1 1 1 1 1 1 1 1 1	0.05

Table 4.2: Haplotypes and their estimated frequencies for the 10 loci dataset

4.2.2 Frequency Estimation

We have examined the accuracy of our method and compared it against HIPPO and HAPLOPOOL on the three datasets described in our previous subsection. In all experiments considered in this subsection the DNA pools were simulated assuming no missing data or measurement error. The performance of the methods is shown in Figure 4.1.

For the 3 and 10 loci datasets the result presented is the average χ^2 distance from a 100 simulation experiments, whereas in the HapMap dataset the result presented is the average χ^2 distance on the 80 datasets considered. For the 3 loci dataset it can be seen that TDSPool and HAPLOPOOL produced similar accuracy. For the remaining two datasets with larger number of loci TDSPool demonstrated superior performance. For the HapMap dataset only TDSPool and HAPLOPOOL were evaluated since the maximum number of loci HIPPO can handle without prior knowledge of the major haplotypes in the population is 10. At the same time even though HAPLOPOOL can in principle handle larger datasets, due to excessive computational time for datasets with 24 and 28 loci we restricted our comparisons to datasets between 10 and 20 loci. We note here as well that since HIPPO is based on a central limit theorem it is likely to be a better approximation in large

pools as opposed to small ones that we focus in our study.

From our experiments we can also see that the number of pools also affected accuracy. All algorithms demonstrated improved performance with increasing number of pools in the dataset.

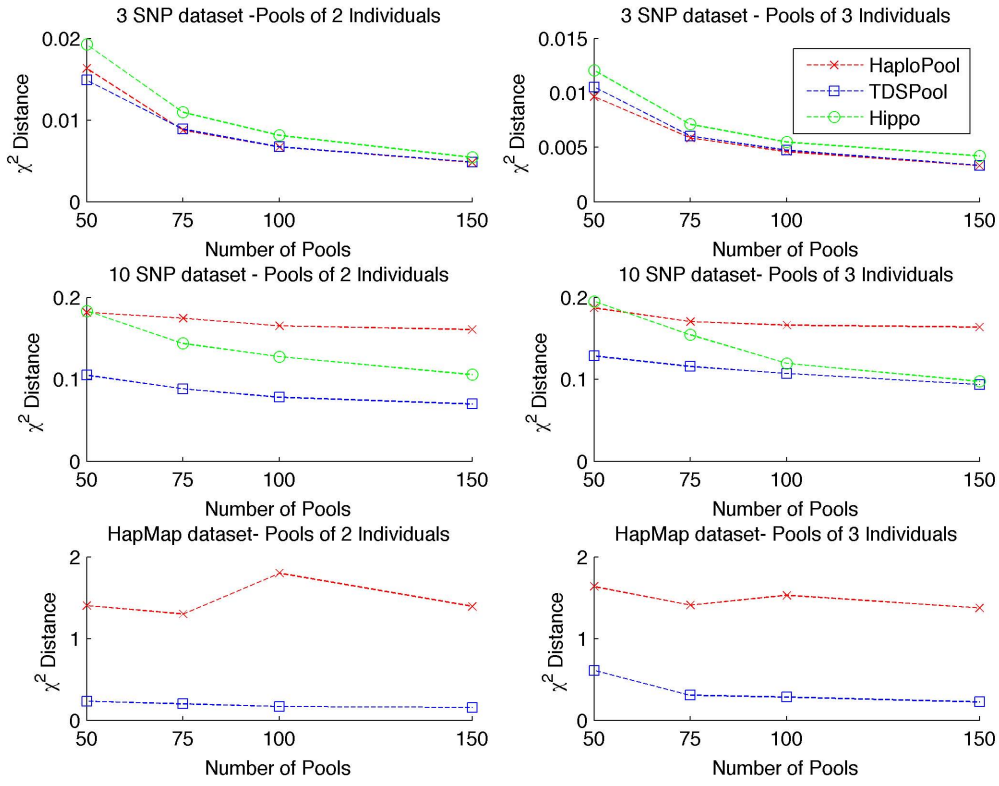

Figure 4.1: Accuracy of haplotype frequency estimates. Estimating χ^2 distance for 3 loci, 10 loci and HapMap dataset for 50, 75, 100 and 150 pools with HAPLOPOOL, TDSPool and HIPPO.

4.2.3 Noise and Missing Data

In the previous subsection we have evaluated the performance of our method by simulating DNA pools without missing data and measurement errors. However, in allelotyping pooled DNA, allele frequencies may not be estimated properly in some practical situations and the data are consequently missing or have measurement errors.

In order to measure the effect of genotype error on the accuracy of the haplotype frequency estimation and evaluate the performance of our method under such scenarios, we have simulated genotyping error by adding a Gaussian error with SD σ to each called allele frequency. Suppose we denote the correct allele frequency at SNP j in pool i as c_{ij}. The perturbed allele frequency is given by $\hat{c_{ij}} = c_{ij} + x$ where $x \sim N(0, \sigma^2)$. After simulating these perturbed haplotype frequencies, we discretize the resulting frequencies to produce perturbed allele counts that are consistent with the number of haplotypes in each pool. We have considered a variety of values for σ, ranging from 0 to 0.06 similar to Kirkpatrik et al. [60]. The perturbed datasets examined were derived from the unperturbed datasets used in the previous subsection with the procedure described above. The results are shown in Figure 4.2. We give the results only when the number of pools is 75 but the shape of the figures is similar for the remaining number of pools examined in our previous subsection.

For small number of loci, HAPLOPOOL achieves the best performance. However, for larger datasets TDSPool outperforms all competing methods.

Furthermore, we have evaluated the performance of our methodology using missing data. We have randomly masked 1 and 2% of the SNPs respectively on the 10 loci datasets and estimated the accuracy. As shown in Figure 4.3, missing SNPs result in small loses in the accuracy and as expected the error decreases with increasing pool number.

4.2.4 Timing Results

The computational times for all datasets are displayed in Table 4.3. All methods were run with their default parameters. Specifically, for HIPPO the default number of iterations was 100000 and

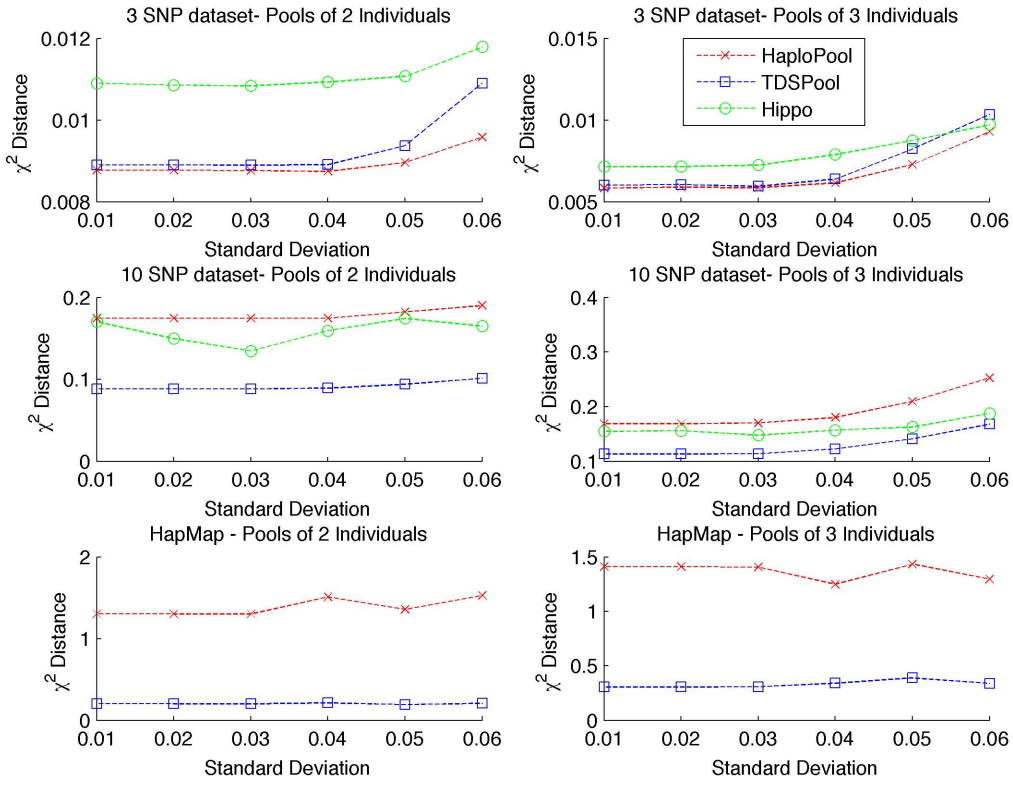

Figure 4.2: Accuracy of haplotype frequency estimates with genotyping errors. Estimating χ^2 distance for 3 loci, 10 loci and HapMap datasets when noise is added on the pooled allele frequencies.

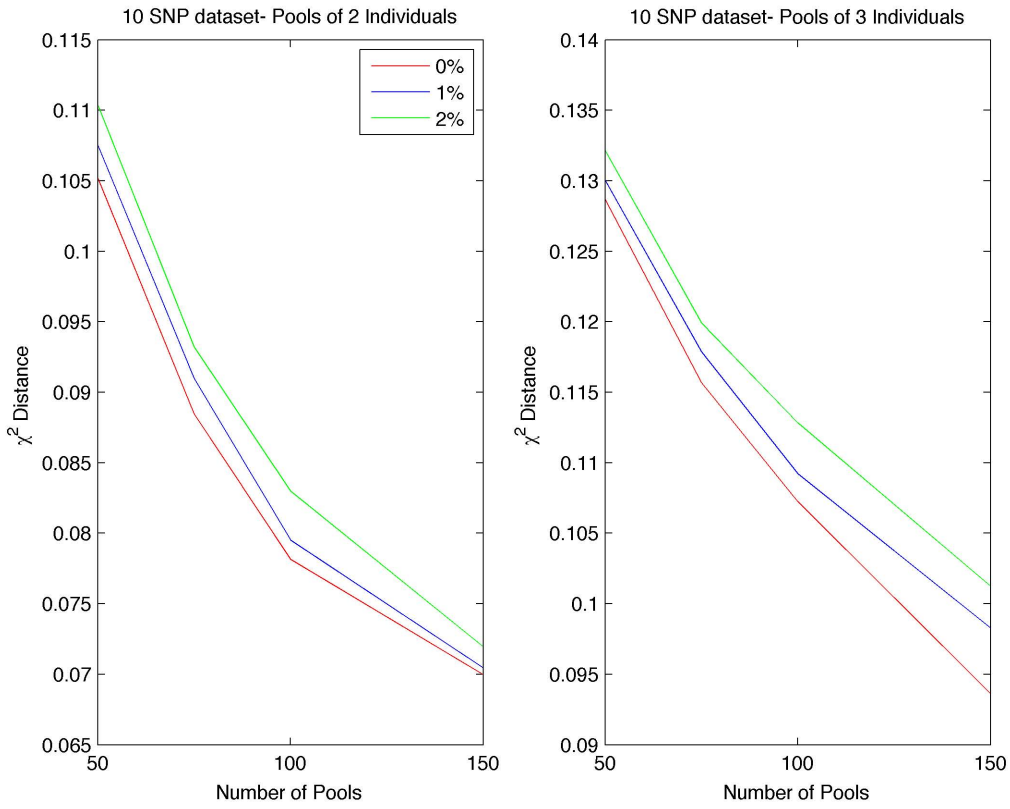

Figure 4.3: Accuracy of haplotype frequency estimates with missing data. Estimating χ^2 distance for 10 loci dataset with 0,1 and 2% of missing SNPs..

for TDSPool the default number of streams (as will be defined in the ”Methods” section) used throughout our experiments was chosen to be 50. Based on these results HIPPO was the slowest performing method in all datasets performing more than 20 times slower than the remaining two algorithms in the ten loci dataset. For the three loci dataset all methods were able to estimate the haplotype frequencies within six seconds. For the ten loci dataset HAPLOPOOL and TDSPool were still able to produce the results in less than three seconds whereas HIPPO demanded more than 58 seconds to finish. For the HapMap datasets again both methods TDSPool and HAPLOPOOL were able to finish the procedure within four seconds. In the ten loci and HapMap datasets TDSPool demonstrated better performance compared to HAPLOPOOL when the number of pools in each dataset was more than 75. Therefore, for all practical applications all methods are fast enough and within limits for researchers to use.

4.3 Discussion

In this chapter, we have introduced a new algorithm for estimating haplotype frequencies from datasets with pooled DNA samples and we have compared it with existing available packages. We have shown that for datasets with small number of loci our algorithm has comparable performance to state-of-the-art methods in terms of accuracy and computational time but it demonstrates superior performance for datasets with larger number of loci.

Our method specifically focuses on small pool sizes and we have demonstrated the performance on pools of two or three individuals. In our experiments we have partitioned pooled genotype vectors in blocks of 4 SNPs as described in the ”Partition-Ligation” subsection. We have chosen to partition the pooled genotypes every 4 SNPs so that computations are performed fast and we avoid

		Number of pools			
		50	75	100	150
3-loci Dataset					
	TDSPool	0.4458	0.4331	0.4743	0.4861
		0.4260	0.4772	0.5346	0.5350
	HaploPool	0.0697	0.0642	0.0607	0.0674
		0.0593	0.0681	0.0607	0.0691
	HIPPO	2.3593	3.0793	3.8856	5.3911
		2.4182	3.2047	4.1161	5.5873
10-loci Dataset					
	TDSPool	0.8094	0.7778	1.0367	1.1259
		1.0269	1.0805	1.1804	1.392
	HaploPool	0.5136	0.7381	0.9554	1.4012
		0.8531	1.2331	1.6247	2.4078
	HIPPO	59.5605	62.7163	64.1563	71.0505
		58.8816	64.6515	64.5386	73.9019
HapMap Dataset					
	TDSPool	1.0189	1.1660	1.1765	1.5455
		1.8760	2.0830	2.1848	3.2719
	HaploPool	0.6737	0.9577	1.2679	1.8489
		1.1636	1.6928	2.2006	3.2905

Table 4.3: Timing Results. For each dataset in each algorithm the first line corresponds to the case that each pool has 2 individuals whereas the second line to the case that each pool has three individuals. Time is given in seconds.

cases with huge number of solutions. Partitioning the dataset every 3 SNPs had negligible impact on the accuracy of our results (results not shown) whereas partitioning every 5 SNPs in general can produce block pool genotypes with thousands of solutions, especially when missing data occur.

In the framework developed by Pirinen, which had resulted in HIPPO, the algorithm was able to accommodate prior database information on existing haplotypes in a population. Similarly, our methodology offers a framework that can easily incorporate prior knowledge in the form of known haplotypes from the same population as that from which the target pools were created. When such existing haplotypes are known (such as those available from the HapMap), they can be easily introduced in the form of a prior for the counts in the TDSPool algorithm. The presence of the extra information will improve the frequency estimation accuracy in the target population.

4.4 Conclusions

We have introduced a new algorithm for estimating haplotype frequencies from pooled DNA samples using a Tree-Based Deterministic sampling scheme. Algorithms for haplotype frequency estimation from pooled data fall into two categories. The first category consists of algorithms that focus on accurate solutions and allow for considerably large genotype segments and the second category of algorithms that focus on small segments but allow for a large number of individuals per pool. We have compared our methodology with state-of-the-art algorithms from each category, namely HAPLOPOOL and HIPPO. We have focused on scenarios and datasets in which the use of pooling data is suggested for haplotype frequency estimation according to the study of Kuk et al. [65] . In specific, our method focuses on scenarios where pools contain 2 or 3 individuals and we have shown that for such scenarios our method demonstrates comparable or better performance

compared with competing algorithms for a small number of loci and outperforms these algorithms for a large number of loci. Furthermore, our TDSPool methodology provides a straightforward framework for incorporating prior database knowledge into the haplotype frequency estimation.

4.5 Methods

In the beginning of the section we introduce some notation. For consistency we develop the adjusted statistical framework for the pooling data. We first present the prior and posterior distribution given the data and then derive the state update equations for the TDSPool estimator. We further present the modified partition-ligation procedure adjusted for the pooled data so that we are able to handle larger haplotype vectors and we finally give a summary of the proposed procedure.

4.5.1 Definitions and Notation

Suppose we are given a set of pooled DNA measurements on L diallelic loci. We denote the two alleles at each locus by 0 and 1, for convenience of our representation. Following the common notation, we use the counts of allele 1 as the measurement for each allele on each pooled DNA sample, which can be converted from the estimated allele frequencies and consists the pool genotype. Therefore if the size of a pool is N individuals, the counts for each allele can vary between 0 and $2N$.

Suppose that we have T such pools each one of them with size $N_j, j = 1,\ldots,T$. We denote $a_t = \{a_t^1,\ldots,a_t^L\}$ to be the pool genotype of the t^{th} pool where $a_t^i \in \{0,\ldots,2N_t\}$. Suppose also that $A_t = \{a_1,\ldots,a_t\}$ is a set of pool genotypes of pools up to and including pool t and let A denote the full set of pool genotypes. In pool t we denote the haplotypes occurring in that pool

as $h_t = \{h_{t,1}, \ldots, h_{t,2N_t}\}$ where $h_{t,i} \in \{0,1\}^L$ is a binary string of length L and the minor allele is present in position j in haplotype i if $h_{t,i,j} = 0$. We further define $H_t = \{h_1, \ldots, h_t\}$, similarly to A_t as the set of haplotypes for each pool genotype up to and including pool t. A schematic representation of the dataset and the notation used is given in Figure 4.4.

Let us also define $Z = \{z_1, \ldots, z_M\}$, where $z_m \in \{0,1\}^L$ is a binary string of length L in which 0 and 1 correspond to the two alleles at each locus, as the set containing all haplotype vectors of length L that are consistent with any pool genotype in the set A. To obtain Z from the given dataset A, we first enumerate for each a_i the subset $\psi_i = \{h_i^1, \ldots, h_i^Y\}, i = 1, \ldots, T$ that contains all possible haplotype assignments which are consistent with a_i. The set Z is then given simply by $\bigcup_{i=1}^{T} \psi_i$. A set of population haplotype frequencies $\theta = \{\theta_1, ..., \theta_M\}$ is also associated with the set Z of all possible haplotype vectors, where θ_m is the probability with which the haplotype z_m occurs in the total population.

4.5.2 Probabilistic model

Similarly to the previous chapters, if we assume random mating in the population it is clear that the number of each unique haplotype in H is drawn from a multinomial distribution based on the haplotype frequency θ which leads us to the use of the Dirichlet distribution as the prior distribution for θ. Similarly, to the previous chapters calculating the posterior distribution for θ we have:

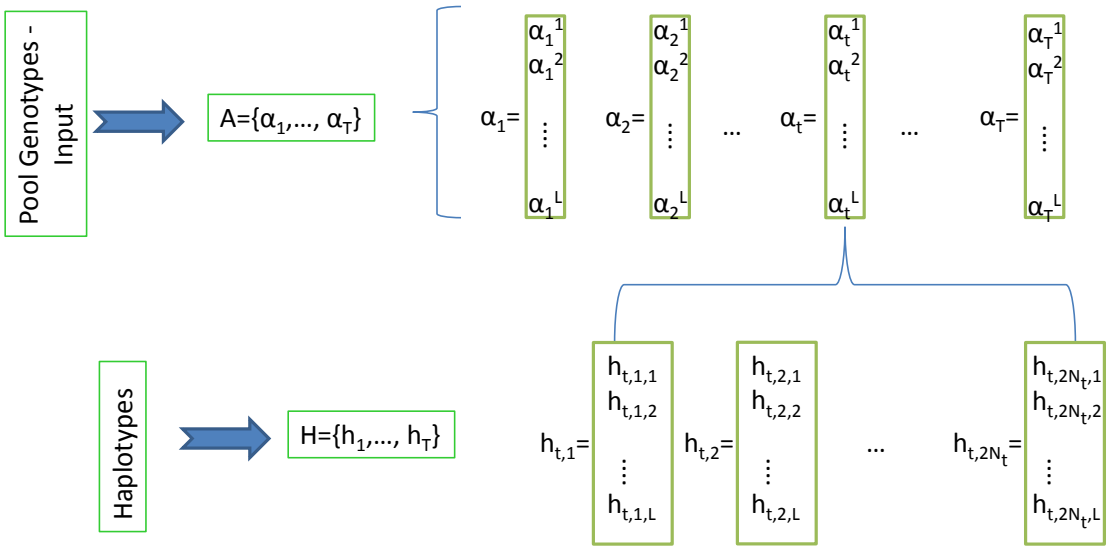

Figure 4.4: Schematic representation of the notation used in our methodology. For each pool genotype (a_t) and at each locus, the value of the pool genotype at that locus a_t^j is the sum of the values on that loci across all haplotypes in that pool i.e. $a_t^j = \sum_{i=1}^{2N_t} h_{t,i,j}$.

$$
\begin{aligned}
& p(\theta|A_t, H_t, Z) \\
& \propto p(a_t|h_t = (h_{t,1}, \ldots, h_{2N_t}), \theta, A_{t-1}, H_{t-1}) \\
& xp(h_t = (h_{t,1}, \ldots, h_{2N_t})|\theta, A_{t-1}, H_{t-1}, Z) p(\theta|A_{t-1}, H_{t-1}, Z) \\
& \propto p(h_t = (h_{t,1}, \ldots, h_{2N_t})|\theta, Z) p(\theta|A_{t-1}, H_{t-1}, Z) \\
& \propto \prod_{i=1}^{2N_t} \theta_{h_{t,i}} \prod_{m=1}^{M} \theta_m^{\rho_m(t-1)-1} \\
& \propto \prod_{m=1}^{M} \theta_m^{\rho_m(t-1)-1+\sum_{i=1}^{2N_t} I(z_m - h_{t,i})} \\
& \propto D(\rho_1(t-1) + \sum_{i=1}^{2N_t} I(z_1 - h_{t,i}), \ldots, \rho_M(t-1) + \sum_{i=1}^{2N_t} I(z_M - h_{t,i}))
\end{aligned}
\tag{4.1}
$$

where we denote $\rho_m(t) m = 1, \ldots, M$ as the parameters of the distribution of θ after the t^{th} pool and $I(z_m - h_{t,i})$ with $i = 1, \ldots, 2N_t$ is the indicator function which equals 1 when $z_m - h_{t,i}$ is a vector of zeros, and 0 otherwise.

Similar to the previous chapters we have obtained that the posterior distribution for θ is also Dirichlet with parameters as given in 4.1 and depends only on the sufficient statistics, $T_t = \{\rho_m(t), 1 \leq m \leq M\}$ which can be easily updated based on T_{t-1} ,h_t, a_t as given by $T_t = T_t(T_{t-1}, h_t, a_t)$.

4.5.3 Inference problem

Following the notation we used in the previous subsections we can summarize the frequency estimation problem as follows: Given $A = \{a_1, \ldots, a_T\}$ the set of observed pool genotype vectors and $Z = \{z_1, \ldots, z_M\}$ the set of haplotypes compatible to the pool genotypes in A we wish to infer $H = \{h_1, \ldots, h_T\}$ the unknown haplotypes in each pool and $\theta = \{\theta_1, ..., \theta_M\}$ the haplotype

frequencies of all the haplotypes occuring in the population.

4.5.4 Computational algorithm (TDSPool)

Similar to traditional Sequential Monte Carlo (SMC) methods, we assume that by the time we have processed pool genotype a_{t-1}, we have K sets of solution streams (i.e. sets of candidate haplotypes for pools $1,\ldots,t-1$) and their associated weights $\{H_{t-1}^{(k)}|w_{t-1}^{(k)}, k=1,\ldots,K\}$ properly weighted with respect to their posterior distribution $p_\theta(H_{t-1}|A_{t-1})$. Given the set of solution streams and the associated weights we approximate the distribution $p(H_{t-1}|A_{t-1})$ as follows:

$$\hat{p}_\theta(H_{t-1}|A_{t-1}) = \frac{1}{w_{t-1}} \sum_{k=1}^{K} w_{t-1}^{(k)} I(H_{t-1} - H_{t-1}^{(k)}) \tag{4.2}$$

where $w_{t-1} = \sum_{k=1}^{K} w_{t-1}^{(k)}$ and $I(*)$ is the indicator function such that $I(x-y) = 1$ for $x = y$ and $I(x-y) = 0$ otherwise.

When we process the pool genotype t we would like to make an online inference of the haplotypes H_t based on the pool genotypes A_t. Let us further assume that there are K^{ext} possible haplotype solutions compatible with the genotype of the t^{th} pool, i.e. $h_t^i, i = 1,\ldots,K^{ext}$.

Before we move to the derivation of the state update equation we note here that in the following we will use the fact that for the unknown parameters θ, as we have shown in "Probabilistic Model" subsection, under certain assumptions the prior and posterior distribution are Dirichlet and depend only on a set of sufficient statistics $T_t = T_t(T_{t-1}, h_t, a_t)$.

Therefore, from Bayes' theorem we have:

$$
\begin{aligned}
p_\theta(H_t|A_t,Z) &\propto p_\theta(a_t|H_t,A_{t-1})p_\theta(h_t|H_{t-1},A_{t-1},Z)p_\theta(H_{t-1}|A_{t-1},Z) \\
&\propto p_\theta(H_{t-1}|A_{t-1},Z)p_\theta(a_t|H_t,A_{t-1})\int p(h_t|H_{t-1},\theta,Z)p(\theta|T_{t-1},Z)d\theta \\
&\propto p_\theta(H_{t-1}|A_{t-1},Z)\int p(h_t|H_{t-1},\theta,Z)p(\theta|T_{t-1},Z)d\theta \\
&\propto p_\theta(H_{t-1}|A_{t-1},Z)\int (\prod_{k=1}^{M}\theta_k^{\sum_{i=1}^{2N_t}I(z_k-h_{t,i})})p(\theta|T_{t-1},Z)d\theta \\
&\propto p_\theta(H_{t-1}|A_{t-1},Z)\int (\prod_{k=1}^{M}\theta_k^{r_k})\frac{1}{B(\rho(t-1))}\prod_{i=1}^{M}\theta_i^{\rho_i(t-1)-1}d\theta \\
&\propto p_\theta(H_{t-1}|A_{t-1},Z)\frac{B(\rho(t-1)+r)}{B(\rho(t-1))}\int \frac{1}{B(\rho(t-1)+r)}\prod_{i=1}^{M}\theta_i^{\rho_i(t-1)+r_i-1}d\theta \\
&\propto p_\theta(H_{t-1}|A_{t-1},Z)\frac{B(\rho(t-1)+r)}{B(\rho(t-1))}
\end{aligned}
\tag{4.3}
$$

where $r=[\sum_{i=1}^{2N_t}I(z_1-h_{t,i}),\ldots,\sum_{i=1}^{2N_t}I(z_M-h_{t,i})]$ and $B(\rho(t-1))=\frac{\prod_{i=1}^{M}\Gamma(\rho_i(t-1))}{\Gamma(\sum_{i=1}^{M}\rho_i(t-1))}$.

Assuming that we have approximated $p(H_{t-1}|A_{t-1})$ as in 4.2, we can approximate $p(H_t|A_t)$ using 4.3 as

$$
\hat{p}_\theta(H_t|A_t)=\frac{1}{w_t^{ext}}\sum_{k=1}^{K}\sum_{i=1}^{K^{ext}}w_t^{(k,i)}I(H_t-[H_{t-1}^{(k)},(h_{t,1}^i,\ldots,h_{t,2N_t}^i)]) \tag{4.4}
$$

The weight update formula is given by

$$
w_t^{(k,j)}\propto w_{t-1}^{(k)}\frac{B(\rho^{(k)}(t-1)+r)}{B(\rho^{(k)}(t-1))} \tag{4.5}
$$

where again $r=[\sum_{i=1}^{2N_t}I(z_1-h_{t,i}^j),\ldots,\sum_{i=1}^{2N_t}I(z_M-h_{t,i}^j)]$ and $\rho^{(k)}(t-1)$ is the parameter vector of the assumed Dirichlet prior which represents how many times we have encountered each haplotype in stream k in the solutions up to pool $t-1$.

4.5.5 Partition-Ligation

In the partition phase the dataset is divided into small segments of consecutive loci. Once the blocks are phased, they are ligated together using a modified extension of the Partition-Ligation (PL) method for the case of pooled data.

In our current implementation to be able to derive all possible solution combinations for each pool genotype efficiently we have decided to keep the maximum block length to 4 SNPs. Clearly the more SNPs are included in a block the more information about the LD patterns we can capture but at the same time the number of possible combinations increases and becomes prohibitive for more than 5 SNPs. For our experiments in a dataset with L loci we have considered $L/4$ blocks of 4 consecutive loci and the remaining SNPs were treated as a separate block.

The result of phasing for each block is a set of haplotype solutions for each pool genotype. Two neighbouring blocks are ligated by creating merged solutions for each pool genotype from all combinations of the block solutions, one from each block. When creating a merged solution for a pool genotype from the two separate solutions (one from each block), since we do not know which haplotypes belong to the same chromosome, all different possible assignments are examined. The TDSPool algorithm is then repeated in the same manner as it was for the individual blocks.

Furthermore, the order in which the individual blocks are ligated is not predetermined. We first ligate the blocks that would produce in each step the minimum entropy ligation. This procedure allows us to ligate first the most homogeneous blocks so that we have more certainty in the solutions that we produce while moving in the ligation procedure.

4.5.6 Summary of the proposed algorithm

Routine 1

- Set the current number of streams $m = 1$. Define K as the maximum number of streams allowed. Define $H_0^1 = \{\}$.
- For $t = 1, 2, \ldots$
  - Find the K^{ext} possible haplotype configurations compatible with the pool genotype of the t^{th} pool.
  - For $k = 1, \ldots, m, j = 1, \ldots, K^{ext}$
    - Enumerate all possible stream extensions $H_t^{(k,j)} = [H_{t-1}^{(k)}, (h_{t,1}^j, \ldots, h_{t,2N_t}^j)]$
    - $\forall j$ compute the weights $w_t^{(k,j)}$ according to 4.5
  - Select and preserve $M = min(K, mK^{ext})$ distinct sample streams $\{H_t^{(k)}, k = 1, \ldots, M\}$ with the highest importance weights $\{w_t^{(k)}, k = 1, \ldots, M\}$ from the set $\{H_t^{(k,j)}, w_t^{(k,j)}, k = 1, \ldots, m, j = 1, \ldots, K^{ext}\}$
  - Update the number of counts of each encountered haplotype in each stream
  - Set $m = M$

TDSPool Algorithm

- Partition the genotype dataset G into B subsets.
- For $b = 1, \ldots, B$ apply Routine 1 so that all segments are phased and for each one keep all the solutions contained in the top K particles

- Until all blocks are ligated, repeat the following
  - Find the blocks that if ligated would produce the minimum entropy
  - Ligate the blocks, following the procedure described in the Partition-Ligation subsection

Chapter 5

Haplotype Inference in Copy Number Variation / SNP data

5.1 Introduction

Copy number variations (CNVs) are a form of a structural genomic variation referring to duplications and deletions of DNA segments larger than 1 kilobase in size. CNVs are abundant in the human genome and it is estimated that they can occupy as much as $4-6\%$ [68–70].

Recently, large-scale genome wide studies have shed light in many aspects and characteristics of CNVs providing unique insights into the origins, mechanisms, formation and population genetics of CNVs [68, 69, 71]. At the same time, CNVs have been associated with complex traits unexplained by recent GWAS [68] and are believed to make a substantial contribution to uncovering the mechanisms and etiology of disease phenotypes that result from complex patterns of inheritance [68, 72].

A variety of techniques exist for CNV detection. Initially, experimental studies have been

performed primarily by array CGH but lately due to improved resolution and genome coverage of genotyping arrays a number of methods have been developed relying on whole-genome SNP genotyping arrays which offer a more sensitive approach and are more suitable for high resolution CNV detection [69,73,74]. As a result there is currently simultaneously information on the integer Copy Number (CN) genotypes along a CNV region and on SNPs outside these regions, in which we will refer in the following as CNV-SNP genotypes.

For diploid organisms theoretical and empirical arguments have been made for the use of haplotypes as opposed to genotypes as discussed in the previous chapters and a variety of methods have been developed. However, only recently this problem has drawn attention when haplotypes are inferred in a CNV-SNP region.

If we focus within a specific CNV region in a sample of individuals and assume that the ploidy is fixed for each individual along the region then the problem of inferring the haplotypes is identical to the problem of inferring the haplotypes in polyploid organisms or estimating haplotypes from pooling data. This problem has been discussed in the previous chapter and not surprisingly a number of algorithms originally developed for this setting have been applied to the associated CNV haplotype inference problem described above.

Apart from the previous scenarios a number of methodologies have been specifically developed and tailored for CNV data. Kato et al [75] have developed a methodology MOCSphaser based on the EM algorithm to assign copy numbers in their respective chromosomes in regions that include CN and SNPs. A core limitation of MOCSphaser as described above is that it takes into consideration only the total CN and not the alleles themselves, assigning on each chromosome a raw CN. As a consequence even though it provides information about the total copies on a chromosome, that could be potentially useful, it does not provide information on the diplotypes

themselves.

Another algorithm recently proposed by Kato et al [76] CNVphaser uses an EM approach to perform inference. The core limitation of that method is that the inference is performed within a CNV region and that the ploidy is considered fixed for an individual within the region. To address these problems and thus enabling the phasing of regions where the ploidy of an individual varies along the region and each individual can have different breakpoints Su et al. [77] suggested polyHap(v2.0) in which they extended the functionality of their original methodology for pooling data [78]. In their study they discern the phasing within a CNV into non-internal phasing in which the CNV in a chromosome is inferred as a diplotype and internal phasing in which the specific haplotypes comprising the CNV in a chromosome are further identified. We will use these definitions in our current work.

In their algorithm Su et al. use an HMM methodology that has separate emission states for the internal and non-internal phasing. They treat the transition between states conceptually in a hierarchical two level model where the first level is for the transition among CN states and the second for the transition among the haplotype states given the CN states. polyHap(v2.0) is the only currently available method that can phase complex CNV regions by allowing arbitrary changes of CN within individuals and along the genomic sequence.

In this chapter, we introduce a related new Sequential Monte Carlo algorithm for haplotype phasing of CNV-SNP data (TDSCNV). In our method samples are processed sequentially and our method scales linearly with the number of samples as well as the number of individuals. We demonstrate that using our methodology we can achieve state-of-the-art performance while our method is an order of magnitude faster than polyHap(v2.0).

5.2 Results

5.2.1 Measurement of Phasing Accuracy

We have used a number of different measures to evaluate the performance of our methodology. First, the switch error rate, as defined in previous chapters, is the percentage of switches among all possible switches in haplotype orientation used to recover the correct phase in an individual. In the case of a small number of loci where haplotype vectors can be expected to be reconstructed exactly we have used two figures of merit, namely the χ^2 and l_1 distance to evaluate the accuracy of frequency estimation. Suppose that f are the predicted haplotype frequencies from an algorithm and g are the gold-standard population level haplotype frequencies. The χ^2 distance between the two distributions as defined in the previous chapter is $\chi^2(f,g) = \sum_{i=1}^{d} (f_i - g_i)^2/g_i$, where g is the expected distribution and d is the number of gold standard haplotypes whereas the l_1 distance between the two distributions is defined as $l_1(f,g) = \sum_{i=1}^{d} |f_i - g_i|$ [60].

5.2.2 Switch Error Rate

We have compared the performance of our method with polyHap(v2.0) for haplotypic phase inference using the switch error rate. In this section the evaluation was done on non-internal haplotypes. In the evaluation of the switch error rate we consider only CN and SNP positions that are ambiguous. For a marker genotype to have ambiguous phasing there should be at least two alternative orientation assignments. As an example all 3CN genotypes are ambiguous positions. This is easy to see, as the choice alone of the chromosome that would have the duplication creates two distinct possible assignments. The performance of our method is shown in Table 5.1. We have considered

	Number of markers		
	30	50	100
TDSCNV	0.111	0.127	0.14
polyHap(v2.0)	0.124	0.135	0.138

Table 5.1: Switch Error rates for non-internal phasing. The switch error rate presented for each number of markers is the average on 100 datasets

three marker sizes namely 30, 50 and 100 markers. For each marker size we have simulated 100 datasets and the result presented is the average error rate on these 100 datasets. We can see that for 30 and 50 markers our method was marginally better than polyHap(v2.0) whereas for the 100 marker datasets it was marginally worse.

5.2.3 Haplotype Frequency Estimation

We have examined the accuracy of our method and compared it against polyHap(v2.0) on datasets of 8 and 10 markers in which individuals had a fixed ploidy. We have evaluated two appropriate figures of merit as described above, the χ^2 and l_1 distance. We should note here that in order to determine how good frequency estimations with a given method are, a small number of markers should be used. The reason is that for a large number of markers it would be unlikely that the exact same haplotypes would appear or reconstructed with appreciable frequency. Our method produced an average χ^2 and l_1 of 0.32 and 0.41 respectively compared to 0.35 and 0.45 produced by polyHap(v2.0).

5.2.4 Internal Phasing

We have further evaluated the performance of our method using the switch error rate inside duplicated regions. In this subsection the evaluation was done on internal phasing and particularly

	Number of markers		
	30	50	100
TDSCNV	2.1	3.7	5.7
polyHap(v2.0)	262.3	431.5	892.1

Table 5.2: Timing Results. For each method and each marker size the computational time is the average time on the 100 datasets used in the switch error rate calculation. Time is given in seconds.

in duplicated segments of a chromosome as the scope was to detect how good the specific haplotypes comprising the duplicated chromosomal region could be recovered. The switch error rate evaluation within such duplicated regions is exactly the same as the evaluation on a genotype with only SNPs. We have used the same 100 datasets for each of the three dataset sizes, namely 30, 50 and 100 markers, as in the evaluation of the switch error rate for non-internal phasing described in a previous subsection. We found, as expected, that the results were similar irrespectively of the dataset size and the average across all datasets was 0.183.

5.2.5 Timing Results

The computational times for the 30, 50 and 100 marker datasets used for the calculation of the Switch Error rate are displayed in Table 5.2. We can see that TDSCNV is an order of magnitude faster than polyHap(v2.0) for all marker sizes examined.

5.3 Methods

The structure of this section is as follows: In the beginning of the section we introduce some notation that we will use throughout the chapter. In the subsections that follow we present the modified version of our TDS methodology for the case of CNV-SNP data. For completeness, we develop again our framework in detail as presented in the second chapter. We first present some

modelling results for the prior and posterior distributions for the population haplotype frequencies given the observed data. We then present the TDS methodology for the cases of known population frequencies and subsequently extend it to the case of unknown frequencies. In the derivation of the later we use the previously derived results for the prior and posterior distributions for the haplotype frequencies. We end the exposition of our method by deriving the state update equations for the TDSCNV estimator and presenting the modified partition-ligation procedure adjusted for the CNV-SNP dataset scenario. In the end of the section we describe the procedure for creating the datasets which we have used in the Results section to evaluate our methodology.

5.3.1 Definitions and Notation

Suppose we are given a set of CNV-SNP genotypes on L diallelic loci. We denote the two alleles at each locus by 0 and 1. In the following we will use the counts of allele 1 as the provided measurement for each allele on each sample. In our method we allow in a specific position a single amplification or deletion. Therefore, if we are within a CNV region in a chromosome the allele counts could range from 0 to 2 but could range from 0 to 1 outside these regions.

Suppose that we have T individuals and we denote $c_t = \{c_t^1, \ldots, c_t^L\}$ to be the observed genotype of the t^{th} sample where $c_t^i \in \{0,1,2,3,4\}$ are the observed counts on the i^{th} position. Suppose also that $C_t = \{c_1, \ldots, c_t\}$ is a set of individuals up to and including individual t and let C denote the full set of individuals.

In terms of haplotypes we make an initial distinction in the values that alleles take in internal and non-internal phasing. The framework that follows however will be described generically and will be the same in both cases.

For non-internal phasing our purpose is to infer haplotypic phase on diploid chromosomes as we are interested in the total copies of an allele at a specific position on a chromosome. Therefore the possible values for an allele at each position are $\{-,0,1,01,00,11\}$. On the contrary for internal phasing we infer haplotypic phase on polyploid chromosomes and the possible alleles at each position are $\{-,0,1\}$.

For individual t we denote the haplotypes occurring in that individual as h_t. In the case of non-internal phasing $h_t = \{h_{t,1}, h_{t,2}\}$. For internal phasing $h_t = \{h_{t,1}, \ldots, h_{t,p}\}$, where p is the ploidy of the organism and $p \in \{1,2,3,4\}$ as in our methodology we only consider a single deletion or a single amplification per chromosome. Therefore, for the case of non-internal phasing $h_{t,1}$, $h_{t,2}$ are strings of length L in which $h_{t,i,j} \in \{-,0,1,01,00,11\}$ and for internal phasing $h_{t,i}$ are strings of length L in which $h_{t,i,j} \in \{-,0,1\}$.

We further denote $H_t = \{h_1, \ldots, h_t\}$, similarly to C_t as the set of haplotypes for each individual up to and including individual t. Let us also define $Z = \{z_1, \ldots, z_M\}$, as the set containing all haplotype vectors of length L that are consistent with any genotype in the set C. To obtain Z from the given dataset C, we first enumerate for each c_i the subset $\psi_i = \{h_i^1, \ldots, h_i^Y\} i = 1, \ldots, T$ that contains all possible haplotype assignments which are consistent with c_i. The set Z is then given simply as $\bigcup_{i=1}^{T} \psi_i$. A set of population haplotype frequencies $\theta = \{\theta_1, ..., \theta_M\}$ is also associated with the set Z of all possible haplotype vectors, where θ_m is the probability with which the haplotype z_m occurs in the total population. We note here once again that we have given the definitions of Z and θ generically for both internal and not internal phasing.

5.3.2 Prior and Posterior Distribution for θ

Simlarly to the previous chapters, if we assume random mating in the population it is clear that the number of each unique haplotype in H is drawn from a multinomial distribution based on the haplotype frequency θ, which leads us to the use of the Dirichlet distribution as the prior distribution for θ so that $\theta \sim D(\rho_1, \ldots, \rho_M)$. It is well known in Bayesian statistics that the Dirichlet distribution is the conjugate prior of the multinomial distribution. This implies in our case, that, if we assume that the prior distribution for θ is Dirichlet and we draw haplotypes based on their frequencies (multinomial distribution), then the posterior distribution for θ is again a Dirichlet distribution. We have proven this fact in previous chapter but for consistency we give a short proof below

$$
\begin{aligned}
& p(\theta|C_t, H_t, Z) \\
& \propto p(c_t|h_t = (h_{t,1}, \ldots, h_p), \theta, C_{t-1}, H_{t-1}) \\
& xp(h_t = (h_{t,1}, \ldots, h_p)|\theta, C_{t-1}, H_{t-1}, Z) p(\theta|A_{t-1}, H_{t-1}, Z) \\
& \propto p(h_t = (h_{t,1}, \ldots, h_p)|\theta, Z) p(\theta|C_{t-1}, H_{t-1}, Z) \\
& \propto \prod_{i=1}^{p} \theta_{h_{t,i}} \prod_{m=1}^{M} \theta_m^{\rho_m(t-1)-1} \\
& \propto \prod_{m=1}^{M} \theta_m^{\rho_m(t-1)-1+\sum_{i=1}^{p} I(z_m - h_{t,i})} \\
& \propto D(\rho_1(t-1) + \sum_{i=1}^{p} I(z_1 - h_{t,i}), \ldots, \rho_M(t-1) + \sum_{i=1}^{p} I(z_M - h_{t,i}))
\end{aligned}
\tag{5.1}
$$

where we denote $\rho_m(t) m = 1, \ldots, M$ as the parameters of the distribution of θ after the t^{th} pool and $I(z_m - h_{t,i})$ with $i = 1, \ldots, 2N_t$ is the indicator function which equals 1 when $z_m - h_{t,i}$ is a vector

of zeros, and 0 otherwise. We note here once again that the number of haplotypes (i.e. the index p in the assignment) depends on the phasing and is 2 for non-internal phasing while it ranges for internal phasing. Furthermore, in the previous calculations for θ for each genotype vector we only consider haplotype configurations that are consistent with that genotype.

We have shown that the posterior distribution for θ is also Dirichlet with parameters as given in 5.1 and depends only on the sufficient statistics, $T_t = \{\rho_m(t), 1 \leq m \leq M\}$ which can be easily updated based on T_{t-1} ,h_t, c_t as given by 5.1 i.e. $T_t = T_t(T_{t-1}, h_t, c_t)$.

5.3.3 TDS Estimator with known frequencies θ

Similar to the derivation in the second chapter, assume that by the time we have processed genotype c_{t-1} we have a set of K potential solution streams $H_{t-1}^{(k)} k = 1, \ldots, K$ each associated with its corresponding weight $\{H_{t-1}^{(k)} | w_{t-1}^{(k)}, k = 1, \ldots, K\}$. At $t-1$ we approximate the real continuous distribution $p(H_{t-1}|G_{t-1})$ as a discrete distribution as follows:

$$\hat{p}_\theta(H_{t-1}|G_{t-1}) = \frac{1}{w_{t-1}} \sum_{k=1}^{K} w_{t-1}^{(k)} I(H_{t-1} - H_{t-1}^{(k)}) \tag{5.2}$$

where $w_{t-1} = \sum_{k=1}^{K} w_{t-1}^{(k)}$ and $I(*)$ is the indicator function such that $I(x-y) = 1$ for $x = y$ and $I(x-y) = 0$ otherwise.

Processing the next individual t we would like to make an online inference of the haplotypes H_t based on the genotypes C_t. From Bayes' theorem we have:

$$\begin{aligned} p_\theta(H_t|C_t) &\propto p_\theta(c_t|H_t, C_{t-1}) p_\theta(H_t|C_{t-1}) \\ &\propto p_\theta(c_t|H_t, C_{t-1}) p_\theta(h_t|H_{t-1}, C_{t-1}) p_\theta(H_{t-1}|C_{t-1}) \end{aligned} \tag{5.3}$$

where for our purposes we only consider haplotype assignments for individual t that are compatible to its observed genotype. Assume further there are K^{ext} such assignments. From previous relationships, if we knew the system parameters θ we would be able to approximate the distribution of $p(H_t|C_t)$ as

$$\hat{p}_\theta(H_t|C_t) = \frac{1}{w_t^{ext}} \sum_{k=1}^{K} \sum_{i=1}^{K^{ext}} w_t^{(k,i)} I(H_t - [H_{t-1}^{(k)}, h_t^{(i)}])$$

where $[H_{t-1}^{(k)}, h_t^{(i)}]$ represents the vector obtained by appending the element $h_t^{(i)}$ to the vector $H_{t-1}^{(k)}$ and $w_t^{ext} = \sum_{i,k} w_t^{(k,i)}$ with

$$w_t^{(k,i)} \propto w_{t-1}^{(k)} p_\theta(c_t|h_t = i) p_\theta(h_t = i|H_{t-1}^{(k)})$$

5.3.4 TDS Estimator with unknown frequencies θ

However, the frequencies θ are not known. In our model we use a Dirichlet distribution, as the prior for θ and as shown we obtain a posterior distribution for θ (given H_t and C_t) that is Dirichlet and only depends on a set of sufficient statistics. Using Bayes' theorem and similarly to the previous subsection we have:

$$
\begin{aligned}
p_\theta(H_t|C_t,Z) &\propto p_\theta(c_t|H_t,C_{t-1})p_\theta(h_t|H_{t-1},C_{t-1})p_\theta(H_{t-1}|C_{t-1},Z) \\
&\propto p_\theta(H_{t-1}|C_{t-1},Z)p_\theta(c_t|H_t,C_{t-1})\int p(h_t|H_{t-1},\theta,Z)p(\theta|T_{t-1},Z)d\theta \\
&\propto p_\theta(H_{t-1}|C_{t-1},Z)\int p(h_t|H_{t-1},\theta,Z)p(\theta|T_{t-1},Z)d\theta
\end{aligned} \tag{5.4}
$$

where again we only consider haplotype assignments that are compatible with the observed genotype.

Taking into consideration as argued before that, if we know the systems parameters θ then the $p(h_t|H_{t-1},\theta,Z)$ terms represents sampling from a multinomial distribution and that the mean of the Dirichlet distribution with respect to an element θ_k of the vector θ is:

$$
E\{\theta_k\} = \frac{\rho_k}{\sum_{j=1}^{M}\rho_j}
$$

we have from 5.4 that:

$$
\begin{aligned}
p_\theta(H_t|C_t,Z) &\propto p_\theta(H_{t-1}|C_{t-1},Z)\int p(h_t|H_{t-1},\theta,Z)p(\theta|T_{t-1},Z)d\theta \\
&\propto p_\theta(H_{t-1}|C_{t-1},Z)\int (\prod_{k=1}^{M}\theta_k^{\sum_{i=1}^{p} I(z_k-h_{t,i})})p(\theta|T_{t-1},Z)d\theta \\
&\propto p_\theta(H_{t-1}|C_{t-1},Z)\int (\prod_{k=1}^{M}\theta_k^{r_k})\frac{1}{B(\rho(t-1))}\prod_{i=1}^{M}\theta_i^{\rho_i(t-1)-1}d\theta \\
&\propto p_\theta(H_{t-1}|C_{t-1},Z)\frac{B(\rho(t-1)+r)}{B(\rho(t-1))}\int \frac{1}{B(\rho(t-1)+r)}\prod_{i=1}^{M}\theta_i^{\rho_i(t-1)+r_i-1}d\theta \\
&\propto p_\theta(H_{t-1}|C_{t-1},Z)\frac{B(\rho(t-1)+r)}{B(\rho(t-1))}
\end{aligned} \tag{5.5}
$$

where $r = [\sum_{i=1}^{p} I(z_1 - h_{t,i}), \ldots, \sum_{i=1}^{p} I(z_M - h_{t,i})]$ and $B(\rho(t-1)) = \frac{\prod_{i=1}^{M} \Gamma(\rho_i(t-1))}{\Gamma(\sum_{i=1}^{M} \rho_i(t-1))}$.

Assuming that we have approximated $p(H_{t-1}|C_{t-1})$ as in 5.2 we can approximate $p(H_t|C_t)$ using 5.5 as

$$\hat{p}_\theta(H_t|C_t) = \frac{1}{w_t^{ext}} \sum_{k=1}^{K} \sum_{i=1}^{K^{ext}} w_t^{(k,i)} I(H_t - [H_{t-1}^{(k)}, (h_{t,1}^i, \ldots, h_{t,p}^i)]) \tag{5.6}$$

where the weight update formula is given by

$$w_t^{(k,j)} \propto w_{t-1}^{(k)} \frac{B(\rho^{(k)}(t-1) + r)}{B(\rho^{(k)}(t-1))} \tag{5.7}$$

where again $r = [\sum_{i=1}^{p} I(z_1 - h_{t,i}^j), \ldots, \sum_{i=1}^{p} I(z_M - h_{t,i}^j)]$ and $\rho^{(k)}(t-1)$ is the parameter vector of the assumed Dirichlet prior which represents how many times we have encountered each haplotype in stream k in the solutions up to individual $t-1$.

5.3.5 Partition-Ligation

In the partition phase the dataset is divided into small segments of consecutive loci and each of the individual blocks is phased separately. To ligate the individual blocks we have adjusted the original Partition-Ligation method for the case of CNV-SNP data.

Similarly to the previous chapter, in our current implementation to be able to derive all possible solution combinations for each pool genotype efficiently we have decided to keep the maximum block length to 5 SNPs. As was the case with the pooling data as well, the more SNPs are included in a block the more information about the LD patterns we can capture but at the same time the number of possible combinations increases and becomes prohibitive for more than 5 SNPs. For

our experiments in a dataset with L loci we have considered $L/5$ blocks of 5 consecutive loci and the remaining SNPs were treated as a separate block.

The result of phasing for each block is a set of haplotype solutions for each genotype. Two neighbouring blocks are ligated by creating merged solutions for each genotype from combinations of the block solutions, each associated with the product of the individual solution weights called the ligation weight.

Depending on which haplotypes one from each block are going to be assigned on the same chromosome for each individual, a different number of changes in the ploidy of that individual will occur. In our method we consider only the assignments that will produce the minimum number of such changes. Therefore, if both haplotypes in any block have the same CN we examine both alternative assignments but we otherwise ligate solutions that have the same CN. The TDS algorithm is then repeated in the same manner as it was for the individual blocks with the weights of the solutions scaled by the associated ligation weight for that solution.

5.3.6 Dataset Creation

Our datasets consisted of SNPs from Chromosomes 1 and 2 from HapMap CEU population (HapMap3 release 2 - Phasing data). For our purposes we have considered only the parents in each trio which are the unrelated individuals in our dataset thus resulting in a total of 88 individuals. We have initially filtered out SNPs with minor allele frequencies less than 5% and we have then considered non-overlapping datasets with a fixed number of SNPs. To create artificial CNV regions within each dataset we have used the following procedure.

First, in each dataset we have found all the different haplotypes appearing in the dataset. In

order to retain as much of the LD structure and also the property that most of the CNVs could be flagged by neighbouring SNPs [68] we have randomly replaced specific areas of randomly chosen haplotypes with a CNV haplotype. To perform that procedure we randomly selected haplotypes based on their frequency in the population and modified them inserting CNV regions sequentially as follows. Each position was considered as the beginning of a CNV region with a probability of 0.1. For each position flagging the beginning of a CNV we assigned the length of the CNV region uniformly between three to eight SNPs. We then progressed along the haplotype from the end of the CNV region in a similar fashion until we reached the end of a given haplotype.

Chapter 6

Conclusions and Future Work

In this thesis we have presented a Sequential Monte Carlo framework (TDS) and tailored it to address instances of haplotype inference and frequency estimation problems. In Chapter 2 we introduced our framework and adjusted it to perform haplotype inference in trio families. We showed that our approach demonstrates a great tradeoff between speed and accuracy making it ideal for routine use.

In Chapter 3 we have extended the applicability of our framework to handle general nuclear families. Our methodology is currently the only framework that simultaneously uses familial and population-based information in studies including nuclear families with any number of children. At the same time we have demonstrated why such approaches are preferable as opposed to resorting to a pedigree based method that would use only the familial constraints or a population based method that would use only the population based information.

In Chapter 4 we have used our TDS framework to address the haplotype inference and frequency estimation problem in pooling data and polyploid organisms. We demonstrated that our method achieves improved performance over existing approaches in datasets with large number of

markers.

In Chapter 5 we have addressed the general inference problem in regions of CNV/SNP data. Our framework enables the phasing of regions where the ploidy of an individual varies along the region and each individual can have different breakpoints.

We have shown that our approach consists a convenient statistical framework able to encompass and efficiently process prior information for the samples. At the same time, its computational performance makes it an ideal option for current genome wide studies.

Apart from the scenarios addressed in this thesis there are other inference scenarios that are considered open problems where our approach could potentially offer a relatively simple and efficient solution and these are discussed below.

6.1 Haplotype inference in datasets that include pedigrees

In Chapters 2 and 3 we have presented our framework for haplotype inference in trio families and subsequently extended it to handle nuclear families in such a way so that it could simultaneously exploit population and familial information.

A clear extension of that scenario is for the case where instead of nuclear families we are presented with general pedigrees. In the nuclear families case, as examined in Chapter 3 there were obvious trio scenarios, that even though suboptimal and with potential drawbacks, could nevertheless consist a viable alternative for a study. However, if pedigrees are included in the dataset it is less than obvious what such scenarios for phasing the pedigrees would be, as they should support a Mendelian consistent allele flow through the pedigree and could probably vary depending on the specific focus of the study.

As discussed in the background chapter of the thesis and the introduction of Chapter 3 a number of methods have been proposed that can perform the inference in isolated pedigrees. These methods can be further decomposed in likelihood based methods and ruled based algorithms. Likelihood based algorithms reconstruct configurations by maximizing the likelihoods or conditional probabilities of the configurations. Rule based algorithms reconstruct configurations by minimizing the total number of recombination events.

The approach we have introduced in Chapter 3 can be seen as a mixture of a population based approach and a rule based approach as in order to reconstruct locally potential solutions for each family we use a minimum recombinant criterion.

A similar two stage approach can be potentially followed in this more general setting. A rule based algorithm can be applied as a first step in each pedigree separately, resulting to a set of potential configurations implying the minimum number of recombination events in that pedigree. We have to note here that each such configuration should include the potential assignment of the founders in the pedigree (individuals with no parents in the pedigree). These configurations would consist the list of potential assignments in the specific pedigree. The MRH algorithm [43] is a potential algorithm that performs such inference using a set of rules. Our statistical framework can then be applied to take advantage of the population information, resolve the ambiguities and perform the inference.

This approach is a viable and novel approach for a small number of markers. However, to be able to extend it to a larger number of markers the partition-ligation method should be used so that we could combine chromosomal segments in the founders. An obvious problem of that approach is that, as was the case in Chapter 3, we may not end up with minimum recombinant orientations in each pedigree. To be more specific, assume that a set of minimum recombinant solutions is kept

for each pedigree in the dataset. Merged solutions from adjacent blocks should then be created by creating combinations of solutions one from each block for each founder in each pedigree, which are then subjected to the TDS procedure. This process does not in principle guarantee that the derived final solution will be a minimum recombinant solution across the entire dataset. This is clear if we think that since the TDS procedure in each block is independent of the adjacent blocks, potential minimum recombinant configurations may be discarded within each block during the TDS scheme. It is important that a detailed evaluation should be done as in Chapter 3 to define what are the potential gains from such an approach.

A more general scenario could occur if individuals in a pedigree or even some of the founders originate from different populations with known database information. This scenario however is uncommon as it could suffer from potential population stratification problems in its consequent downstream association analysis. In association studies samples are taken from the same population in order to avoid population stratification problems that could arise from such structures. Nevertheless, it could still be the case that some distant members in a pedigree may originate from different ethnical groups and different options should be evaluated for these members in the analysis.

6.2 Haplotype inference in CNV/SNP regions in nuclear families

As presented in the introduction of the previous chapter a number of methods for CNV detection rely on whole genome SNP genotyping arrays. As a result there is currently simultaneously infor-

mation on the integer Copy Number along a CNV region and on SNPs outside these regions for which we have used the term CNV/SNP regions.

As explained in the introduction of Chapters 2 and 3 in many of these studies trio information or information in nuclear families is available. In addition, in many studies such information is further desired so that the computational inference of the boundaries of CNV regions can be more accurate. The problem of inferring CNV/SNP haplotypes in trio or nuclear families becomes therefore relevant.

In these scenarios the presence of the familial information will impose constraints on the set of possible haplotype orientations for the parents similar to the cases examined in the first chapters of this thesis where familial information is also available. The constraints are similar for both cases of internal and non-internal phasing. For the regions were no CN exists in an individual, the problem is exactly the same as the ones examined in Chapters 2 and 3.

Bibliography

[1] A. G. Clark, "The role of haplotypes in candidate gene studies," *Genet Epidemiol*, vol. 27, pp. 321–33, Dec 2004.

[2] D. E. Weeks, E. Sobel, J. R. O'Connell, and K. Lange, "Computer programs for multilocus haplotyping of general pedigrees," *Am J Hum Genet*, vol. 56, pp. 1506–7, Jun 1995.

[3] G. Gao, D. B. Allison, and I. Hoeschele, "Haplotyping methods for pedigrees," *Hum Hered*, vol. 67, no. 4, pp. 248–66, 2009.

[4] D. J. Schaid, "Evaluating associations of haplotypes with traits," *Genet Epidemiol*, vol. 27, pp. 348–64, Dec 2004.

[5] C. Durrant, K. T. Zondervan, L. R. Cardon, S. Hunt, P. Deloukas, and A. P. Morris, "Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes," *Am J Hum Genet*, vol. 75, pp. 35–43, Jul 2004.

[6] J. Liu, C. Papasian, and H.-W. Deng, "Incorporating single-locus tests into haplotype cladistic analysis in case-control studies," *PLoS Genet*, vol. 3, p. e46, Mar 2007.

[7] S.-Y. Su, D. J. Balding, and L. J. M. Coin, "Disease association tests by inferring ancestral haplotypes using a hidden markov model," *Bioinformatics*, vol. 24, pp. 972–8, Apr 2008.

[8] A. P. Morris, J. C. Whittaker, and D. J. Balding, "Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies," *Am J Hum Genet*, vol. 70, pp. 686–707, Mar 2002.

[9] B. Rannala and J. P. Reeve, "High-resolution multipoint linkage-disequilibrium mapping in the context of a human genome sequence," *Am J Hum Genet*, vol. 69, pp. 159–78, Jul 2001.

[10] S. Zöllner and J. K. Pritchard, "Coalescent-based association mapping and fine mapping of complex trait loci," *Genetics*, vol. 169, pp. 1071–92, Feb 2005.

[11] C. Durrant and A. P. Morris, "Linkage disequilibrium mapping via cladistic analysis of phase-unknown genotypes and inferred haplotypes in the genetic analysis workshop 14 simulated data," *BMC Genet*, vol. 6 Suppl 1, p. S100, 2005.

[12] B. L. Browning and S. R. Browning, "Efficient multilocus association testing for whole genome association studies using localized haplotype clustering," *Genet Epidemiol*, vol. 31, pp. 365–75, Jul 2007.

[13] P. C. Sabeti, D. E. Reich, J. M. Higgins, H. Z. P. Levine, D. J. Richter, S. F. Schaffner, S. B. Gabriel, J. V. Platko, N. J. Patterson, G. J. McDonald, H. C. Ackerman, S. J. Campbell, D. Altshuler, R. Cooper, D. Kwiatkowski, R. Ward, and E. S. Lander, "Detecting recent positive selection in the human genome from haplotype structure," *Nature*, vol. 419, pp. 832–7, Oct 2002.

[14] P. Fearnhead and P. Donnelly, "Estimating recombination rates from population genetic data," *Genetics*, vol. 159, pp. 1299–318, Nov 2001.

[15] S. R. Myers and R. C. Griffiths, "Bounds on the minimum number of recombination events in a sample history," *Genetics*, vol. 163, pp. 375–94, Jan 2003.

[16] M. Bahlo and R. C. Griffiths, "Inference from gene trees in a subdivided population," *Theor Popul Biol*, vol. 57, pp. 79–95, Mar 2000.

[17] P. Beerli and J. Felsenstein, "Maximum likelihood estimation of a migration matrix and effective population sizes in n subpopulations by using a coalescent approach," *Proc Natl Acad Sci U S A*, vol. 98, pp. 4563–8, Apr 2001.

[18] N. Papadopoulos, F. S. Leach, K. W. Kinzler, and B. Vogelstein, "Monoallelic mutation analysis (mama) for identifying germline mutations," *Nat Genet*, vol. 11, pp. 99–102, Sep 1995.

[19] D. J. Schaid, "Relative efficiency of ambiguous vs. directly measured haplotype frequencies," *Genet Epidemiol*, vol. 23, pp. 426–43, Nov 2002.

[20] J. A. Douglas, M. Boehnke, E. Gillanders, J. M. Trent, and S. B. Gruber, "Experimentally-derived haplotypes substantially increase the efficiency of linkage disequilibrium studies," *Nat Genet*, vol. 28, pp. 361–4, Aug 2001.

[21] J. Marchini, D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. S. Qin, H. M. Munro, G. R. Abecasis, P. Donnelly, and International HapMap Consortium, "A comparison of phasing algorithms for trios and unrelated individuals," *Am J Hum Genet*, vol. 78, pp. 437–50, Mar 2006.

[22] "http://hapmap.ncbi.nlm.nih.gov/."

[23] M. Stephens and P. Scheet, "Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation," *Am J Hum Genet*, vol. 76, pp. 449–62, Mar 2005.

[24] D. Brinza and A. Zelikovsky, "2snp: scalable phasing method for trios and unrelated individuals," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 5, no. 2, pp. 313–8, 2008.

[25] E. Halperin and E. Eskin, "Haplotype reconstruction from genotype data using imperfect phylogeny," *Bioinformatics*, vol. 20, pp. 1842–9, Aug 2004.

[26] S. Lin, A. Chakravarti, and D. J. Cutler, "Haplotype and missing data inference in nuclear families," *Genome Res*, vol. 14, pp. 1624–32, Aug 2004.

[27] Z. S. Qin, T. Niu, and J. S. Liu, "Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms," *Am J Hum Genet*, vol. 71, pp. 1242–7, Nov 2002.

[28] T. Niu, Z. S. Qin, X. Xu, and J. S. Liu, "Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms," *Am J Hum Genet*, vol. 70, pp. 157–69, Jan 2002.

[29] S. R. Browning and B. L. Browning, "Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering," *Am J Hum Genet*, vol. 81, pp. 1084–97, Nov 2007.

[30] B. L. Browning and S. R. Browning, "A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals," *Am J Hum Genet*, vol. 84, pp. 210–23, Feb 2009.

[31] S. F. Schaffner, C. Foo, S. Gabriel, D. Reich, M. J. Daly, and D. Altshuler, "Calibrating a coalescent simulation of human genome sequence variation," *Genome Res*, vol. 15, pp. 1576–83, Nov 2005.

[32] A. Kong, D. F. Gudbjartsson, J. Sainz, G. M. Jonsdottir, S. A. Gudjonsson, B. Richardsson, S. Sigurdardottir, J. Barnard, B. Hallbeck, G. Masson, A. Shlien, S. T. Palsson, M. L. Frigge, T. E. Thorgeirsson, J. R. Gulcher, and K. Stefansson, "A high-resolution recombination map of the human genome," *Nat Genet*, vol. 31, pp. 241–7, Jul 2002.

[33] L. Excoffier and M. Slatkin, "Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population," *Mol Biol Evol*, vol. 12, pp. 921–7, Sep 1995.

[34] W. X. Liang KC, "A deterministic sequential monte carlo method for haplotype inference," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 322–331, 2008.

[35] A. Iliadis, J. Watkinson, D. Anastassiou, and X. Wang, "A haplotype inference algorithm for trios based on deterministic sampling," *BMC Genet*, vol. 11, p. 78, 2010.

[36] S. Lin, D. J. Cutler, M. E. Zwick, and A. Chakravarti, "Haplotype inference in random population samples," *Am J Hum Genet*, vol. 71, pp. 1129–37, Nov 2002.

[37] E. S. Lander and P. Green, "Construction of multilocus genetic linkage maps in humans," *Proc Natl Acad Sci U S A*, vol. 84, pp. 2363–7, Apr 1987.

[38] G. R. Abecasis, S. S. Cherny, W. O. Cookson, and L. R. Cardon, "Merlin–rapid analysis of dense genetic maps using sparse gene flow trees," *Nat Genet*, vol. 30, pp. 97–101, Jan 2002.

[39] L. Kruglyak, M. J. Daly, M. P. Reeve-Daly, and E. S. Lander, "Parametric and nonparametric linkage analysis: a unified multipoint approach," *Am J Hum Genet*, vol. 58, pp. 1347–63, Jun 1996.

[40] D. F. Gudbjartsson, K. Jonasson, M. L. Frigge, and A. Kong, "Allegro, a new computer program for multipoint linkage analysis," *Nat Genet*, vol. 25, pp. 12–3, May 2000.

[41] D. F. Gudbjartsson, T. Thorvaldsson, A. Kong, G. Gunnarsson, and A. Ingolfsdottir, "Allegro version 2," *Nat Genet*, vol. 37, pp. 1015–6, Oct 2005.

[42] A. L. Williams, D. E. Housman, M. C. Rinard, and D. K. Gifford, "Rapid haplotype inference for nuclear families," *Genome Biol*, vol. 11, no. 10, p. R108, 2010.

[43] D. Qian and L. Beckmann, "Minimum-recombinant haplotyping in pedigrees," *Am J Hum Genet*, vol. 70, pp. 1434–45, Jun 2002.

[44] J. R. O'Connell, "Zero-recombinant haplotyping: applications to fine mapping using snps," *Genet Epidemiol*, vol. 19 Suppl 1, pp. S64–70, 2000.

[45] K. Zhang, F. Sun, and H. Zhao, "Haplore: a program for haplotype reconstruction in general pedigrees without recombination," *Bioinformatics*, vol. 21, pp. 90–103, Jan 2005.

[46] M. Fishelson, N. Dovgolevsky, and D. Geiger, "Maximum likelihood haplotyping for general pedigrees," *Hum Hered*, vol. 59, no. 1, pp. 41–60, 2005.

[47] C. S. Carlson, M. A. Eberle, M. J. Rieder, Q. Yi, L. Kruglyak, and D. A. Nickerson, "Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium," *Am J Hum Genet*, vol. 74, pp. 106–20, Jan 2004.

[48] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. de Bakker, M. J. Daly, and P. C. Sham, "Plink: a tool set for whole-genome association and population-based linkage analyses," *Am J Hum Genet*, vol. 81, pp. 559–75, Sep 2007.

[49] W. M. Brown, J. Pierce, J. E. Hilner, L. H. Perdue, K. Lohman, L. Li, R. B. Venkatesh, S. Hunt, J. C. Mychaleckyj, P. Deloukas, and Type 1 Diabetes Genetics Consortium, "Overview of the mhc fine mapping data," *Diabetes Obes Metab*, vol. 11 Suppl 1, pp. 2–7, Feb 2009.

[50] A. Bansal, D. van den Boom, S. Kammerer, C. Honisch, G. Adam, C. R. Cantor, P. Kleyn, and A. Braun, "Association testing by dna pooling: an effective initial screen," *Proc Natl Acad Sci U S A*, vol. 99, pp. 16871–4, Dec 2002.

[51] L. F. Barcellos, W. Klitz, L. L. Field, R. Tobias, A. M. Bowcock, R. Wilson, M. P. Nelson, J. Nagatomi, and G. Thomson, "Association mapping of disease loci, by use of a pooled dna genomic screen," *Am J Hum Genet*, vol. 61, pp. 734–47, Sep 1997.

[52] N. Norton, N. M. Williams, M. C. O'Donovan, and M. J. Owen, "Dna pooling as a tool for large-scale association studies in complex traits," *Ann Med*, vol. 36, no. 2, pp. 146–52, 2004.

[53] J. V. Pearson, M. J. Huentelman, R. F. Halperin, W. D. Tembe, S. Melquist, N. Homer, M. Brun, S. Szelinger, K. D. Coon, V. L. Zismann, J. A. Webster, T. Beach, S. B. Sando, J. O. Aasly, R. Heun, F. Jessen, H. Kolsch, M. Tsolaki, M. Daniilidou, E. M. Reiman, A. Papassotiropoulos, M. L. Hutton, D. A. Stephan, and D. W. Craig, "Identification of the genetic basis for complex disorders by use of pooling-based genomewide single-nucleotide-polymorphism association studies," *Am J Hum Genet*, vol. 80, pp. 126–39, Jan 2007.

[54] P. Sham, J. S. Bader, I. Craig, M. O'Donovan, and M. Owen, "Dna pooling: a tool for large-scale association studies," *Nat Rev Genet*, vol. 3, pp. 862–71, Nov 2002.

[55] Y. Zuo, G. Zou, and H. Zhao, "Two-stage designs in case-control association analysis," *Genetics*, vol. 173, pp. 1747–60, Jul 2006.

[56] T. Ito, S. Chiku, E. Inoue, M. Tomita, T. Morisaki, H. Morisaki, and N. Kamatani, "Estimation of haplotype frequencies, linkage-disequilibrium measures, and combination of haplotype copies in each pool by use of pooled dna data," *Am J Hum Genet*, vol. 72, pp. 384–98, Feb 2003.

[57] S. Wang, K. K. Kidd, and H. Zhao, "On the use of dna pooling to estimate haplotype frequencies," *Genet Epidemiol*, vol. 24, pp. 74–82, Jan 2003.

[58] Y. Yang, J. Zhang, J. Hoh, F. Matsuda, P. Xu, M. Lathrop, and J. Ott, "Efficiency of single-nucleotide polymorphism haplotype estimation from pooled dna," *Proc Natl Acad Sci U S A*, vol. 100, pp. 7225–30, Jun 2003.

[59] M. Pirinen, S. Kulathinal, D. Gasbarra, and M. J. Sillanpää, "Estimating population haplotype frequencies from pooled dna samples using phase algorithm," *Genet Res (Camb)*, vol. 90, pp. 509–24, Dec 2008.

[60] B. Kirkpatrick, C. S. Armendariz, R. M. Karp, and E. Halperin, "Haplopool: improving haplotype frequency estimation through dna pools and phylogenetic modeling," *Bioinformatics*, vol. 23, pp. 3048–55, Nov 2007.

[61] H. Zhang, H.-C. Yang, and Y. Yang, "Poool: an efficient method for estimating haplotype frequencies from large dna pools," *Bioinformatics*, vol. 24, pp. 1942–8, Sep 2008.

[62] A. Y. C. Kuk, H. Zhang, and Y. Yang, "Computationally feasible estimation of haplotype frequencies from pooled dna with and without hardy-weinberg equilibrium," *Bioinformatics*, vol. 25, pp. 379–86, Feb 2009.

[63] D. Gasbarra, S. Kulathinal, M. Pirinen, and M. J. Sillanpää, "Estimating haplotype frequencies by combining data from large dna pools with database information," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 8, no. 1, pp. 36–44, 2011.

[64] M. Pirinen, "Estimating population haplotype frequencies from pooled snp data using incomplete database information," *Bioinformatics*, vol. 25, pp. 3296–302, Dec 2009.

[65] A. Y. C. Kuk, J. Xu, and Y. Yang, "A study of the efficiency of pooling in haplotype estimation," *Bioinformatics*, vol. 26, pp. 2556–63, Oct 2010.

[66] B. J. Barratt, F. Payne, H. E. Rance, S. Nutland, J. A. Todd, and D. G. Clayton, "Identification of the sources of error in allele frequency estimations from pooled dna indicates an optimal experimental design," *Ann Hum Genet*, vol. 66, pp. 393–405, Nov 2002.

[67] S. Jain, X. Tang, C. S. Narayanan, Y. Agarwal, S. M. Peterson, C. D. Brown, J. Ott, and A. Kumar, "Angiotensinogen gene polymorphism at -217 affects basal promoter activity and is associated with hypertension in african-americans," *J Biol Chem*, vol. 277, pp. 36889–96, Sep 2002.

[68] D. F. Conrad, D. Pinto, R. Redon, L. Feuk, O. Gokcumen, Y. Zhang, J. Aerts, T. D. Andrews, C. Barnes, P. Campbell, T. Fitzgerald, M. Hu, C. H. Ihm, K. Kristiansson, D. G. Macarthur, J. R. Macdonald, I. Onyiah, A. W. C. Pang, S. Robson, K. Stirrups, A. Valsesia, K. Walter, J. Wei, Wellcome Trust Case Control Consortium, C. Tyler-Smith, N. P. Carter, C. Lee, S. W. Scherer, and M. E. Hurles, "Origins and functional impact of copy number variation in the human genome," *Nature*, vol. 464, pp. 704–12, Apr 2010.

[69] R. Redon, S. Ishikawa, K. R. Fitch, L. Feuk, G. H. Perry, T. D. Andrews, H. Fiegler, M. H. Shapero, A. R. Carson, W. Chen, E. K. Cho, S. Dallaire, J. L. Freeman, J. R. González, M. Gratacòs, J. Huang, D. Kalaitzopoulos, D. Komura, J. R. MacDonald, C. R. Marshall, R. Mei, L. Montgomery, K. Nishimura, K. Okamura, F. Shen, M. J. Somerville, J. Tchinda, A. Valsesia, C. Woodwark, F. Yang, J. Zhang, T. Zerjal, J. Zhang, L. Armengol, D. F. Conrad, X. Estivill, C. Tyler-Smith, N. P. Carter, H. Aburatani, C. Lee, K. W. Jones, S. W. Scherer, and M. E. Hurles, "Global variation in copy number in the human genome," *Nature*, vol. 444, pp. 444–54, Nov 2006.

[70] M. Kato, S. Yoon, N. Hosono, A. Leotta, J. Sebat, T. Tsunoda, and M. Q. Zhang, "Inferring haplotypes of copy number variations from high-throughput data with uncertainty," *G3 (Bethesda)*, vol. 1, pp. 35–42, Jun 2011.

[71] D. F. Conrad and M. E. Hurles, "The population genetics of structural variation," *Nat Genet*, vol. 39, pp. S30–6, Jul 2007.

[72] S. A. McCarroll and D. M. Altshuler, "Copy-number variation and association studies of human disease," *Nat Genet*, vol. 39, pp. S37–42, Jul 2007.

[73] K. Wang, M. Li, D. Hadley, R. Liu, J. Glessner, S. F. A. Grant, H. Hakonarson, and M. Bucan, "Penncnv: an integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data," *Genome Res*, vol. 17, pp. 1665–74, Nov 2007.

[74] F. J. Steemers and K. L. Gunderson, "Whole genome genotyping technologies on the beadarray platform," *Biotechnol J*, vol. 2, pp. 41–9, Jan 2007.

[75] M. Kato, Y. Nakamura, and T. Tsunoda, "Mocsphaser: a haplotype inference tool from a mixture of copy number variation and single nucleotide polymorphism data," *Bioinformatics*, vol. 24, pp. 1645–6, Jul 2008.

[76] M. Kato, Y. Nakamura, and T. Tsunoda, "An algorithm for inferring complex haplotypes in a region of copy-number variation," *Am J Hum Genet*, vol. 83, pp. 157–69, Aug 2008.

[77] S.-Y. Su, J. E. Asher, M.-R. Jarvelin, P. Froguel, A. I. F. Blakemore, D. J. Balding, and L. J. M. Coin, "Inferring combined cnv/snp haplotypes from genotype data," *Bioinformatics*, vol. 26, pp. 1437–45, Jun 2010.

[78] S.-Y. Su, J. White, D. J. Balding, and L. J. M. Coin, "Inference of haplotypic phase and missing genotypes in polyploid organisms and variable copy number genomic regions," *BMC Bioinformatics*, vol. 9, p. 513, 2008.