Genotype Imputation Methods for Next Generation Datasets

Simone Rubinacci

St Catherine's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2019

Acknowledgements

I am deeply grateful to my DPhil supervisor, Prof. Jonathan Marchini, for his support and encouragement during these years. Without his guidance and constant feedback, especially in the last year, this DPhil would not have been achievable.

I thank Prof. Pier Francesco Palamara for his supervision, support and useful conversations. I would also like to thank Prof. Olivier Delaneau for his guidance during my visit in his lab in Switzerland. That was one of the most valuable experiences I did during my entire DPhil.

I would like to thank all members of the St Catherine's college MCR football, it is sad to leave the captaincy of such a good team, but I am sure I left you in very good hands.

A special thank goes to Iida, you should know that your encouragement and support was worth more than I can express on paper. You always brightened my days when they did not.

Finally, I want to express my very profound gratitude to my parents for providing me with continuous encouragement throughout my DPhil. *Grazie.*

Abstract

The advent of genome-wide association studies (GWAS) revolutionized the field of complex disease genetics. The primary goal of these studies is to achieve a better understanding of the biology of a complex disease and use this knowledge for prevention or better treatment of diseases. Genotype imputation is one of the key steps in such studies. The contribution of genotype imputation is to increase the power of study by boosting the coverage of genetic variation and to provide a natural framework for combining results across association studies that rely on different genotyping platforms.

Genotype imputation is the process of statistical inference of unobserved genotypes in a sample of individuals. In the typical scenario, a reference panel of haplotypes is used to infer ungenotyped variants in a set of study samples. Increasing the sample size in the reference panel improves imputation accuracy, especially for variants with low minor allele frequencies, but big reference panels are also a computational challenge for imputation methods. The goal of this dissertation is to develop methods and protocols that scale genotype imputation to very large next generation reference panels.

The main contribution of this dissertation is a genotype imputation method named IMPUTE5. It achieves fast, accurate, and memory-efficient imputation by selecting a small number of reference panel haplotypes using the Positional Burrows-Wheeler Transform. Imputation is performed only using the selected haplotypes, leading to a dramatic speed-up with no loss in accuracy compared to other methods. In order to facilitate other researchers to perform GWAS, a protocol to impute from a reference panel of phased haplotypes into a genome-wide association dataset is also described.

We also present an efficient C++ implementation of the Positional Burrows-Wheeler Transform which allows fast string matching in a set of haplotypes. An evaluation of previous state selection algorithms is provided together with a qualitative measure of the accuracy of chromosome painting performed by IMPUTE5.

A major application is the imputation of the UK Biobank dataset using the 100,000 Genomes Project reference panel. We show how the reference panel has been created and how imputation will be performed. Imputation of the UK Biobank represents an extremely valuable resource for researches, potentially providing new highlights for genome-wide association studies.

Contents

List of Abbreviations

1000G . . . . . 1000 genomes

AF . . . . . Allele frequency

BCF . . . . . Binary variant call format

BREF3 . . . . Binary reference format version 3

EM . . . . . Expectation-maximization

GEL . . . . . Genomics England Project (alias for 100,000 Genomes Project)

GWAS . . . . . Genome wide association study

HLA . . . . . Human leukocyte antigen

HMM . . . . . Hidden Markov model

HRC . . . . . Haplotype reference consortium

IBD . . . . . Identity by descent

IBS . . . . . Identity by state

IMP5 . . . . . Impute 5 format

LD . . . . . Linkage disequilibrium

MAC . . . . . Minor allele count

MAF . . . . . Minor allele frequency

MCMC . . . . Markov chain Monte-Carlo

PBWT . . . . Positional Burrows-Wheeler transformation

QC . . . . . Quality control

SNP . . . . . Single nucleotide polymorphisms

TOPMed . . . Trans-omics for precision medicine

UKB . . . . . UK Biobank

VCF . . . . . Variant call format

WGS . . . . . Whole genome sequencing

Exploring the unknown
requires tolerating uncertainty.

— Brian Greene

1 Introduction and Background

Contents

1.1 Human genome

The human genome is the set of nucleic acid sequences, encoded in the form of Deoxyribonucleic Acid (DNA), located in the nucleus of each cell. DNA is a double-stranded molecule, organised in 23 pairs of chromosomes in humans, with

each strand composed of a linear sequence of four different nucleotides which are Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). The genome is inherited from an individual's parents. Humans have haploid genomes in egg or sperm cells, created by the meiosis process, and diploid genomes in all other cells.

Individuals in a population show genetic variation. While most of the individuals carry genomes that have the same nucleotide in a given location, called locus, there can be individuals with different nucleotides at the same locus. This nucleotide difference is called a genetic variant. There are two main processes that lead to variability in DNA: recombination and mutation.

During meiosis, crossover of pieces of DNA can recombine together to form a new combination. The expected number of recombinations between two loci is called the genetic distance. Genetic distance increases monotonically with physical distance, but not always linearly, and it is possible to have regions with small genetic distance or regions that exhibit elevated rates of recombination, called recombination hotspots.

Mutation is another process responsible for genetic variation. Mutation refers to the rare event that the genetic material is altered, due to errors during DNA replication or other types of DNA damage. If this change has no effect on cell viability, the mutation can be inherited and eventually spread in the population.

Different nucleotides at a genetic variant are called alleles. Genetic variants can be biallelic, when the variant has only two alleles, and multiallelic, when the variant presents more than two alleles. In this thesis we always refer to biallelic variants, unless otherwise stated.

The incidence of an allele in a population is described by the allele frequency. For a variant, the allele that does occur less frequently in a population is called minor allele. Variants can be grouped by the minor allele frequency (MAF). If MAF $> 1\%$, the variant is called common variant, whereas if MAF $< 1\%$, the variant is called rare variant[1].

[1]The 1% threshold for common and rare variants is quite arbitrary and there is no full agreement on this, but it is usually in the range of [1-0.05]%.

The genotype at a variant is the combination of the two alleles. Let A and B be the two alleles of a variant, where A is considered to be the reference allele in the population and B is the alternative allele. The possible genotypes for the variant are then A/A, A/B and B/B. By counting the number of copies of the alternative allele B, the genotype takes value 0, 1 or 2.

1.1.1 Linkage disequilibrium

A haplotype is defined as a set of consecutive alleles co-localised on the same chromosome. The human genome shows a haplotype block structure, where haplotype blocks span regions of the genome in which there is little evidence of genetic recombination. The low recombination rate in those regions results in a small number of distinct haplotypes per block. Indeed, because of a small number of recombination events, the alleles of an entire haplotype block are likely to be inherited together. Thus, a de novo mutation arising within a haplotype block is also likely to be inherited together with all the nearby alleles of the same block and to be spread to the descendants. However, with the increasing number of generations, the probability that a recombination event cuts down the original segment of alleles containing the mutation also increases. This makes the shared segment smaller and smaller between the descendants and the original ancestor. Hence, the correlation between the variant and its neighbours decreases over time at the population level decreases over time.

Suppose that, among a population, allele 1 occurs with frequency f_k at locus k and allele 1 at locus l has frequency f_l. Similarly, let f_{kl} be the frequency of which allele 1 appears in both locus k and l. The association between the two alleles can be regarded as completely random when the occurrence of one does not affect the occurrence of the other, in which case the probability that both allele 1 occur together in locus k and l is given by the product $f_k f_l$. We say that there is a linkage disequilibrium between the two alleles whenever f_{kl} differs from $f_k f_l$ for any reason. We define the so-called disequilibrium coefficient:

$$D_{kl} = f_{kl} - (f_k f_l) \tag{1.1}$$

which assumes value 0 if the two loci are independent and is also said to be in linkage equilibrium. If the loci are not independent ($D_{kl} \neq 0$), we say that the loci are in linkage disequilibrium (LD). In real datasets, we do not know the exact frequencies in population and we use sample estimates to compute D_{kl}.

We can measure the amount of LD also by using the Pearson's correlation coefficient r^2_{kl}, where r and D are related in the following way:

$$r_{kl} = \frac{D_{kl}}{\sqrt{f_k(1-f_k)f_l(1-f_l)}} \tag{1.2}$$

It follows that two SNPs are in complete LD (not separated by recombination) when $D_{kl} = 1$ or $r^2 = 1$ and in linkage equilibrium when they have both value zero.

1.1.2 Hardy-Weinberg equilibrium

Following the approach suggested by Matti Pirinen[2], let us consider a marker in a population, having allele frequency f_a. Assuming that the two alleles composing a genotype g_i in one individual i are sampled at random from the population, then the genotype frequencies in the population follow the binomial distribution $Bin(2, f_a)$:

$$f_g(g_i) = \begin{cases} (1-f_a)^2, & \text{if } g_i = 0 \\ 2f_a(1-f_a), & \text{if } g_i = 1 \\ f_a^2, & \text{if } g_i = 2 \end{cases} \tag{1.3}$$

These are the Hardy-Weinberg equilibrium (HWE) genotype frequencies. They define a theoretical equilibrium of genotype frequencies given the value of f_a in an ideal randomly-mating population with no selection, migration, or genetic drift.

Most variants across different human populations follow HWE approximately. Deviations from HWE can be due to the influence of recent population structure, assortative mating or natural selection. In addition to this, deviation from HWE can also be due to technical problems in genotype calling due to bad quality data

[2]Lecture notes of the course "Genome-wide Association Studies" by Matti Pirinen, University of Helsinki, 2019.

or presence of a multi-allelic SNP. For this reason, variants that do not follow approximately HWE are usually excluded from genome-wide association studies as part of quality control procedures.

A testing procedure is then required to verify deviations from HWE. For a population of N samples, suppose we have observed genotype counts n_0, n_1, n_2; ($n_0 + n_1 + n_2 = N$), for genotypes 0,1 and 2, respectively, obtaining an allele frequency of $f_a = (n_1+2n_2)/(2N)$. The estimated genotype counts are then $h_0 = N(1-f_a)^2, h_1 = 2Nf_a(1-f_a)$ and $h_2 = Nf_a^2$. We can measure the deviation of the expected counts and the observed counts:

$$t_{\text{HWE}} = \sum_{i=0}^{2} \frac{(n_i - h_i)^2}{h_i} \tag{1.4}$$

and in the case HWE holds at the marker, t_{HWE} is approximately a Chi-square distribution, from which a P-value can be derived. However, for small sample size or rare variants, a test statistic distribution based on permutations should be used instead, since the Chi-square asymptotic approximation is not valid.

1.2 Genome-wide association studies

Genome-wide association studies (GWAS) aim to quantify statistical association between genetic variation and phenotypes in a sample of individuals (Bush and Moore 2012; Tam et al. 2019). The attention of a genome-wide association study is not on Mendelian disorders, caused by a single genetic defect, but on complex diseases, influenced by a combination of multiple genes and environmental factors, such as diabetes or schizophrenia (Mills and Rahal 2019; Cantor et al. 2010).

Typically genome-wide association studies combine data across multiple datasets to analyse millions of variants. Discoveries have confirmed evidence of previously suspected molecular mechanisms (Coon et al. 2007), revealed new pathways (Locke et al. 2015) and have provided insights for the development of new drugs and treatment strategies (Okada et al. 2014; Zhang et al. 2015; Visscher et al. 2017). GWAS have showed that, in general, complex diseases are highly polygenic[3], and

[3]A phenotype is called polygenic if it is influenced by many genes.

many common variants have only small effects on these phenotypes (Visscher et al. 2017). The emergence of these studies has revolutionised the field of complex disease genetics by providing a systematic method that allows researchers to find deeper insights about disease biology.

An example of GWAS on quantitative traits is the GWAS on body-mass index (BMI) (Locke et al. 2015). The study combined data of 339,224 individuals from 125 studies around the world to study genetic association to BMI. The authors found 97 BMI-associated statistically significant loci. Pathway analysis implicates that particular genes and pathways that affect BMI are related to synaptic plasticity and glutamate signalling. These results strengthened the connection between obesity and other metabolic diseases, highlighted new processes and molecular pathways that contribute to obesity and guided further research aimed at unravelling the complex biology of obesity.

There are also some limitations that come with GWAS. Detecting the exact markers that are causal is a problem that cannot be handled with statistical information alone, due to correlation among genetic variants (linkage disequilibrium). This problem, which involves association of signals to causal variants using statistical methods, is called *fine-mapping*.

Another limitation of GWAS is estimating the effect of ultra-rare variants. In order to detect significant association of a rare variant, the sample size or the effect size of the variant must be large (Tam et al. 2019). In addition to that, a typical GWAS is based on genotyping chips which have been designed using linkage disequilibrium. Non-present genotypes in the genotyping chip can be imputed using a reference panel of haplotypes, effectively increasing the number of variants in the study. Imputed data in a marker is, however, dependent on whether the rare allele has been seen in the reference panel. If no minor allele was observed in the reference panel, imputation does not provide any useful information for the marker in question. The progress in sequencing technologies allows now to study the role of rare genetic variants in complex traits using whole-genome sequencing (WGS) data (Kiryluk 2016). However, SNP genotyping and imputation are still orders

of magnitudes cheaper than whole-genome sequencing, making it a cost-effective way to perform GWAS (B. L. Browning, Zhou et al. 2018).

1.2.1 A simple GWAS model

It is widely accepted that contributions of individual SNPs to complex traits are roughly additive (Bush and Moore 2012). This means that the heterozygote risk is intermediate between the two homozygote risks (Balding 2006). To model this, a simple statistical framework to model association between genotypes and a trait is then an additive model, assuming that the means of the effect additively depend on the number of alternative alleles in the genotype[4]. We can then fit a linear model:

$$y = \mu + x\beta + \epsilon \tag{1.5}$$

where y is the phenotype we are observing and x is the genotype assuming values between 0 and 2 (number of alternative alleles copies). The parameters we estimate are μ, the mean of genotype 0, and β, the effect on the mean phenotype per each copy of the alternative allele. The errors are independent, have constant variance and are normally distributed, thus they have identical Normal distribution $N(0, \sigma^2)$ where σ^2 is not known and will be estimated from the data. When we fit the model, we get the parameter estimates $\hat{\mu}$ and $\hat{\beta}$, together with their standard errors.

We are interested in measuring how much variation in y is left unexplained by the model by computing residual sum of squares (RSS) and minimising it:

$$RSS = \sum_i \left(y_i - \hat{\mu} - x_i\hat{\beta}\right) \tag{1.6}$$

The r-squared (r^2) statistic provides a measure of how well the model is fitting the data, taking the form of a proportion of variance. The adjusted version can be expressed as $r^2 = 1 - \frac{RSS}{n-2}/S_{yy}$, where S_{yy} is the sample variance of y, $S_{yy} = \frac{1}{N}\sum_i (y_i - \overline{y})^2$. r^2 is a measure of the linear relationship between our predictor variable and our effect variable. It takes value that lies between 0 and 1, and

[4]We follow the notation as in the lecture notes of the course "Genome-wide Association Studies" by Matti Pirinen, University of Helsinki, 2019

larger values mean more variance explained by the model, i.e. if the model explains data perfectly ($RSS = 0$), then $r^2 = 1$.

The model is a good candidate to test for association, because, even when additivity is not perfectly expressed, one of the three genotypes usually shows much smaller effect, making the additive model still approximately correct. In addition to this, almost all known associations show additive effects, making the additive model better suited for these variants (Hill et al. 2008). Therefore, this simple model is widely used in practice.

1.2.2 Cost-effective GWAS

Whole-genome sequencing can be considered a gold standard for GWAS. Although the price of WGS is constantly falling, GWAS that use sequencing data in large samples are still currently not economically sustainable. However, large projects such as the UK Biobank (Bycroft et al. 2018) have just announced that all the 500,000 participants of the study will be sequenced by the year 2023. This will allow researchers to perform GWAS on a dataset of an unprecedented scale.

In the meantime, GWAS based on SNP arrays, combined with genotype imputation using large WGS reference panels, will be complementary to the study of rare variants and will continue to provide major advances in the field (Tam et al. 2019). In addition, since the price of SNP arrays is also constantly decreasing and recent algorithmic advances lowered the price of imputation to 0.01 US dollars per genotype (B. L. Browning, Zhou et al. 2018), genotype imputation remains as the most cost-effective choice for GWAS.

1.3 The genotype imputation problem

Genotype imputation involves using observed information, in the form of a reference panel of haplotypes at a dense set of SNPs, to infer ungenotyped variants in a study sample (here we refer to the study sample as target panel). In the typical scenario, the study sample is genotyped on a SNP array containing only 300,000 to 5,000,000 variants on a genome-wide scale (LaFramboise 2009), while the reference

panel is high-coverage sequenced, containing even hundreds of millions of markers. Thus, most of the variants in the study samples are missing.

A problem related to genotype imputation is haplotype phasing. Phasing is the process of inferring haplotypes from genotype data[5]. These two problems are linked, since imputation arises naturally in phasing algorithms, and often imputation can be incorporated into phasing methods in order to remove missing data from incomplete datasets. Interestingly, phasing is used for imputation too. A big advance in the computational efficiency of genotype imputation came with the realisation that target individuals could be phased prior to imputation. This process is called pre-phasing (B. Howie et al. 2012).

Usually genotype imputation starts by phasing the study samples, followed by a haploid imputation step in which each haplotype is imputed separately. With the advent of pre-phasing, only haploid imputation is performed. Pre-phasing and haploid imputation is faster and more accurate than diploid imputation (Whalen et al. 2018)

Imputation methods typically rely on a Hidden Markov Model that identifies probabilities of sharing sequences between the haplotypes in the study sample and the haplotypes in the reference panel. The HMM models the haplotype to be imputed as a mosaic of haplotypes in the reference panel and the process of imputation can be seen as an imperfect copy from the haplotypes of the reference panel. The same HMM is also used in other areas of genetics, such as in haplotype phasing or chromosome painting. This is the main reason why there are strong connections between models and methods used to infer haplotype phase and those used to perform genotype imputation (Marchini 2011).

Reference panels have constantly increased their size over time and there are large ongoing sequencing projects that will sequence hundreds of thousands of individuals at high depth in the next few years. These projects will allow us to assemble reference panels of unprecedented scale. In general, it is highly desirable to increase the number of samples and sequencing depth in reference panels in order to improve imputation accuracy and the number of SNPs in the GWAS dataset.

[5]A more detailed description of the haplotype phasing problem is given in Section 1.4

Pre-phasing and the linear forward-backward algorithm on the HMM keep the computation complexity of genotype imputation linear in the number of haplotypes and markers in the reference panel. However, reference panels produced from whole-genome sequencing, such as the Haplotype Reference Consortium (HRC) (McCarthy et al. 2016) have thousands of samples and large numbers of rare variants, and this poses a challenge for imputation methods. This is the reason why the development of new imputation methods is focused on improving the computational complexity from linear to sub-linear in the number of haplotypes of the reference panel.

1.3.1 Uses of genotype imputation

The main application of genotype imputation is facilitating genome-wide association studies in several ways. The benefits of imputation are described below:

Boosted power of GWAS

The first reason of imputation is simply to increase the number of SNPs that can be tested for association in GWAS. This increased number of SNPs boost the power of the study due to an increased chance of the presence of true causal variants in the set of markers tested for association. In addition to this, when missing data is present in the GWAS dataset, imputation effectively increases the sample size for the marker by imputing data for the samples that have missing data. This also leads to an increased power of GWAS.

However, tests of association conducted at millions of imputed loci yields many more (independent) statistical tests. Researchers should be aware to address multiple testing corrections appropriately, in order to avoid false positive results. This is even more problematic in the case of whole genome sequencing given the amount of rare variants in the GWAS cohort. More details about this problem can be found in (Pulit et al. 2017; Gao 2011; M. Li et al. 2012).

Fine-mapping

Imputation provides a more detailed view of associated regions by adding more variants, thereby increasing the chances of identifying a causal variant. For example, in a study made using the Wellcome Trust Case Control Consortium, fine-mapping performed in the region of the known type 2 diabetes gene TCF7L2 showed that the imputed variant *rs7903146* had the strongest study association signal (Marchini, B. Howie et al. 2007).

Meta-analysis

Imputation has been widely used to allow GWAS from cohorts that have been genotyped on different SNP arrays, effectively having information on different sets of markers. Genotype imputation uses a dense reference panel of haplotypes and provides a natural framework to obtain a unique set of SNPs where a genome-wide analysis can be carried. The imputed dataset has the same sample size of the sum of all the samples in the different cohorts and contains the same number of markers as the common reference panel. Several GWAS have been performed using meta-analysis of different cohorts studies (e.g. Lambert et al. 2013; Pharoah et al. 2013).

1.3.2 Li and Stephens model

Several imputation and phasing methods are based on the same statistical framework, the Li and Stephens model (N. Li and Stephens 2003). The original idea was to model linkage disequilibrium, inferring the population-scaled recombination rate ρ, in a tractable framework for large datasets.

Given N haplotypes sampled from a population, $\{h_1, h_2, \ldots, h_N\}$, the original goal of the model was to estimate the likelihood of the parameter ρ.

$$L(\rho) = Pr(h_1, h_2, \ldots, h_N|\rho) = Pr(h_1|\rho)Pr(h_2|h1;\rho), \ldots, Pr(h_N|h_1, h_2, \ldots, h_{N-1};\rho) \tag{1.7}$$

Since the conditional distributions on the right side are hard to compute exactly, the authors have replaced it with approximate distributions $\hat{\pi}$ obtaining the so-called "product of approximate conditionals" (PAC) model:

$$L_{\text{PAC}}(\rho) = \hat{\pi}(h_1|\rho)\hat{\pi}(h_2|h_1;\rho), \ldots, \hat{\pi}(h_N|h_1, h_2, \ldots, h_{N-1};\rho) \tag{1.8}$$

where the quality of the model depends on the distribution $\hat{\pi}$. In the model, the conditional $Pr(h_N|h_1, h_2, \ldots, h_{N-1})$ is calculated using a Hidden Markov Model (HMM) where h_N is seen as an imperfect mosaic of $h_1, h_2, \ldots, h_{N-1}$, meaning that when the distribution of the newly observed haplotype h_N needs to be computed, the other haplotypes $h_1, h_2, \ldots, h_{N-1}$ are taken as the state space of the HMM.

HMMs represent a simple and parsimonious model where observed data is seen to be emitted from the underlying hidden states (Rabiner 1989). There are three canonical problems to solve with HMMs:

- Forward-backward algorithm is used to compute the probability of a particular output sequence.
- Viterbi algorithm is used to find the most likely sequence of (hidden) states which could have generated a given output sequence.
- The Baum-Welch algorithm is used to fit the model parameters, given an output sequence.

Exploiting the independence structure of the Li and Stephens model, due to symmetries in the transition rates, the computation of these three algorithms is very efficient. In the haploid case, computational time is linear with the number of markers and the number of hidden states.

Adopting the Li and Stephens HMM for imputation, each reference haplotype represents a hidden state of the HMM. The model assumes that the target haplotype (at genotyped markers) emerges from an imperfect mosaic of haplotypes in the reference panel, as emitted from a sequence of hidden states. Points of change from one state to another can be interpreted as recombination events and the actual observed alleles may differ from the alleles on the underlying hidden states to allow for mutation and genotype error (Marchini, B. Howie et al. 2007).

1.3.3 Hidden Markov model for imputation

Here we describe a haploid version of the IMPUTE1 model (Marchini, B. Howie et al. 2007). Let H be a set of N haplotypes genotyped at M markers in a reference panel of haplotypes[6]. We also have a set of K study sample (target) haplotypes, defined only at a subset of the M markers. We refer to the set of T markers that are genotyped in both the panels as *target markers* ($\mathcal{T}$), and the others, present only in the reference panel, as *reference markers* ($\mathcal{R}$).

Consecutive pairs of haplotypes represent the diplotype of each study individual. We define the HMM model only at target markers. Therefore, we use the symbol $H^{\mathcal{T}}$ to indicate the restriction of the reference panel H to target markers and use the symbol m to indicate a marker in $\mathcal{T}$.

Given a target haplotype $t = \{t_1, t_2, \ldots, t_T\}$, the probability of observing t from $H^{\mathcal{T}}$ can be then written as:

$$Pr(t|H^{\mathcal{T}}, \rho) = \sum_{Z} Pr(t|Z)Pr(Z|H^{\mathcal{T}}, \rho) \tag{1.9}$$

where Z is a sequence of unobserved copying labels, $Z_m \in \{0, 1, \ldots, N-1\}$, and the term $Pr(Z|H^{\mathcal{T}}, \rho)$ models sequence of transitions of the HMM and is defined by

$$Pr(Z|H^{\mathcal{T}}, \rho) = Pr(Z_1) \prod_{m=1}^{T} Pr(Z_{m+1}|Z_m) \tag{1.10}$$

$$Pr(Z_1 = n) = \frac{1}{N}; \tag{1.11}$$

$$Pr(Z_{m+1} = i|Z_m = j) = \begin{cases} (1-\rho_m) + \frac{\rho_m}{N} & \text{if } i = j, \\ \frac{\rho_m}{N} & \text{otherwise.} \end{cases} \tag{1.12}$$

where $\rho = \{\rho_1, \rho_2, \ldots, \rho_{T-1}\}$ is a locus specific parameter modelling genetic recombination events, defined as $\rho_m = 1 - e^{\frac{-4N_e(r_{m+1}-r_m)}{N}}$, where N_e is the effective diploid population size and $r_{m+1} - r_m$ is the average rate of crossover per unit physical distance per meiosis between target markers $m+1$ and m multiplied by their physical distance. Equation 1.12 is motivated by the fact that switches in the

[6]In this section we use N as the size of a whole reference panel. However, methods such as IMPUTE2 or IMPUTE5 select a subset of these haplotypes to be considered as the actual reference panel. The way different models represent information from a reference panel is not the goal of this section, where a general HMM defined on a set of N haplotypes is presented.

haplotype being copied between m and $m+1$ can be described as a Poisson process having rate $\frac{4N_e(r_{m+1}-r_m)}{N}$ (N. Li and Stephens 2003). The probability of not having recombination events between size m and $m+1$ is then $1-\rho_m$. The probability of having at least one recombination event is ρ_m, which incorporates the probability of having even multiple recombinations occurring between marker m and $m+1$.

The emission probability $Pr(t_m = a|Z_m = n, H^{\mathcal{T}}, \theta)$ has the following form:

$$Pr(t_m = a|Z_m = n) = \begin{cases} \frac{N}{N+\theta} + \frac{\theta}{2(N+\theta)} & \text{if } H^{\mathcal{T}}_{n,m} = a \\ \frac{\theta}{2(N+\theta)} & \text{otherwise.} \end{cases} \tag{1.13}$$

this equation can be explained in a similar way as we did for Equation 1.12. θ has the form of the Watterson coefficient $(\sum_{n=1}^{N-1} \frac{1}{n})^{-1}$.

Imputation of Ungenotyped Variants

For each marker $m \in \mathcal{T}$ the marginal posterior distribution of the copying state can be written as (Marchini 2019):

$$\begin{aligned} Pr(Z_m|t, H, \theta, \rho) &= Pr(Z_m|t_{1:m}, t_{m+1:M}, H, \theta, \rho) \\ \propto \quad Pr(Z_m|t_{1:m}, H^{\mathcal{T}}, \rho, \theta) &Pr(t_{m+1:M}|Z_m, H^{\mathcal{T}}, \rho, \theta) \end{aligned} \tag{1.14}$$

The two terms on the right can be computed using the forward-backward algorithm (Rabiner 1989). The classic Baum-Welch forward equation, indicated by L, defines the probablity of the copying label assuming a particular value, $L(Z_m = n)$, given the first m observations (N. Li and Stephens 2003). It can be expressed recursively as follows:

$$L(Z_1 = n) = \frac{1}{N} Pr(t_1|Z_1 = n); \tag{1.15}$$

$$L(Z_{m+1} = n) = \underbrace{\left((1-\rho_m)L(Z_m = n) + \frac{\rho_m}{N}\sum_{i=1}^{N} L(Z_m = i)\right)}_{\text{transition}} \underbrace{Pr(t_{m+1}|Z_{m+1} = n)}_{\text{emission}} \tag{1.16}$$

where ρ_m, defined in Equation 1.12, denotes the template switch probability between markers m and $m+1$, and $Pr(t_{m+1}|Z_{m+1} = n)$ is the genotype emission probability. A very similar definition can be given for the backward equation.

The most efficient implementations calculate and store these quantities (vectors of length N) at just the $|\mathcal{T}|$ genotyped markers; the distribution of sites in $\mathcal{I}$ are directly calculable from them. Due to the symmetry in the Li and Stephens transition probabilities, the algorithm scales linearly in the number of reference haplotypes and the number of genotype markers $O(N \cdot |\mathcal{T}|)$.

Once the marginal posterior distribution of the copying states at all genotype markers has been computed, the imputation step is performed. Let $r \in \mathcal{R}$ be a marker to impute between two genotype sites m and $m+1$. Then the marginal copying probabilities at r can be linearly interpolated from the posterior distribution at the two neighbouring sites:

$$Pr(Z_r|t, H^{\mathcal{T}}, \theta, \rho) = w_r Pr(Z_m|t, H^{\mathcal{T}}, \rho, \theta) + (1 - w_r) Pr(Z_{m+1}|t, H, \rho, \theta) \quad (1.17)$$

where w_r is defined as:

$$w_r = \frac{g(m+1) - g(r)}{g(m+1) - g(m)} \quad (1.18)$$

and $g(\bullet)$ is the genetic position of the marker (genetic distance). The motivation of using linear interpolation is that, over short genetic distances, the change in state probabilities can be approximated by a straight line.

We can get the marginal posterior distribution of the unobserved alleles at all sites $r \in \mathcal{R}$ as:

$$Pr(t_r = a|H^{\mathcal{T}}, \theta, \rho) = Pr(t_r|Z_r, \theta) Pr(Z_r|t, H^{\mathcal{T}}, \rho, \theta) \quad (1.19)$$

where $a = 0$ or $a = 1$. These allelic probabilities are then combined to output genotype probabilities.

It is crucial that the target and reference datasets are perfectly aligned in order to unambiguously map the same genetic positions. Firstly, the datasets must be aligned in terms of genome build, i.e. both should be represented in build 38, and secondly, they must use the same strand representation of the alleles. This is usually called strand problem. It is required to check the strand of both the datasets, before performing imputation.

1.3.4 Imputation models

IMPUTE1 and IMPUTE2

IMPUTE is a model based on an extension of the Li and Stephens HMM (N. Li and Stephens 2003). IMPUTE1 (Marchini, B. Howie et al. 2007) was one of the first imputation models that has been released and implemented a diploid imputation model. IMPUTE2 (B. N. Howie et al. 2009) introduced the approximation of using only a subset of the reference haplotypes by choosing the best k_{hap} haplotypes in terms of Hamming distance. The HMM is only computed on this subset of haplotypes. IMPUTE2 implements both haploid and diploid imputation.

IMPUTE4

IMPUTE4 (Bycroft et al. 2018) is a re-coded version of the haploid imputation functionality implemented in IMPUTE2 that performs imputation on the whole set of haplotypes of the reference panel.

IMPUTE4 is an efficient implementation that exploits a compact representation of the reference panel by storing the indices of haplotypes carrying the minor allele of markers in the reference panel. This reduces RAM usage and also increases the speed of the computation of the forward-backward probabilities. Recalling Equation 1.19, the advantage of this approach is that, since the model is defined only on biallelic sites and

$$\sum_{n=1}^{N} Pr(Z_r = n|t, H^{\mathcal{T}}, \theta, \rho) = 1 \tag{1.20}$$

this term needs only be calculated for the states $Z_r = n$ where haplotype carries the alternative allele at that site, which can be done efficiently by storing allele indices.

BEAGLE5

BEAGLE5 (B. L. Browning, Zhou et al. 2018) is another model based on the Li and Stephens HMM. BEAGLE5 builds a target-specific set of composite reference haplotypes, where each of them is a mosaic of segments of reference haplotypes. Each segment has a long shared interval with the target haplotype.

These intervals are called identity by state (IBS) segments. This is based on the assumption that each long IBS segment should contain at least one long identity by descendent (IBD) segment. A relatively small number of composite reference haplotypes can be used with large marker windows, as each composite reference haplotype contains several long IBS segments.

BEAGLE5 also collapses closely linked target markers within a small window (by default 0.005 cM) into a single aggregate marker. The forward-backward calculations are only computed on these aggregate sites, reducing the amount of calculations.

Another difference between the IMPUTE model and BEAGLE5 is that the emission probabilities are defined in terms of a user-specified allele error rate. This type of emission model is a generalisation of other methods such as MACH (Y. Li et al. 2010). However, it has been shown that imputation accuracy is relatively insensitive to the emission model and the error rate parameter (B. L. Browning and S. R. Browning 2016).

BEAGLE5 uses two computational shortcuts to reduce the computational time required to estimate the imputed allele probabilities in Equation 1.17. The first is based on the fact that, by compressing reference haplotypes using unique sequences in small windows, the computation of Equation 1.17 is the same for all reference haplotypes sharing a particular sequence. Consequently, when estimating allele probabilities at an imputed marker r, it is only required to sum over the number of distinct allele sequences in the reference panel between aggregate genotyped markers m and $m+1$. The second computational shortcut is the use of a threshold to run imputation only on using highly probable states and discard states that have sufficiently small probability (B. L. Browning and S. R. Browning 2016).

BREF3 file format Another improvement of BEAGLE5 is the development of the BREF3 file format. The total time spent for imputation might be dominated by the time spent reading the reference panel, especially in the case of a big reference panel and a small set of target haplotypes. BEAGLE5 proposes a binary reference

panel file format that is very quick to read in memory and has all the information and data structures used by BEAGLE5 internally already pre-computed.

In the BREF3 format, chromosomes are broken into consecutive, non-overlapping intervals. If the non-major alleles of a marker are rare, BREF3 format stores the indices of the haplotypes that carry each non-major allele. For the remaining markers, BREF3 stores the distinct allele sequences and a pointer from each haplotype to the unique sequence carried by the haplotype in a certain interval. This reduces memory requirements, as the number of distinct allele sequences is typically much less than the number of haplotypes.

Reading BREF3 file with BEAGLE5 is very efficient. In our experiments, BEAGLE5 takes ≈ 2 seconds to read a reference panel containing 200,000 haplotypes and 747,162 markers, approximately ≈ 400 times faster than the time used to read a gzipped-compressed VCF file of the same reference panel.

A single reference panel is used to impute several GWAS datasets, for example in the context of an imputation server. Generating a BREF3 file format is a one-time pre-processing step that increases the efficiency of several subsequent imputation tasks using the same reference panel.

MINIMAC3

MINIMAC3 (Das et al. 2016) is an imputation method that uses a compressed representation on the reference panel; similarly to BEAGLE, the reference panel is split into consecutive segments of sites having a single overlapping site[7].

The forward-backward equations can be expressed in the form of the reduced space of the unique sequences. In a certain block, the original $H = \{h_1, h_2, \ldots, h_N\}$ haplotypes can be compressed using the distinct unique haplotypes as $Y = \{y_1, y_2, \ldots, y_U\}$, where $U \leq N$ and typically, in a small segment, $U \ll N$. Let N_u be the number of haplotypes in H matching y_u in the block.

[7]Another unpublished version called MINIMAC4 has been released. It is more efficient than MINIMAC3 and introduces new parameters to control automated chunking and the use of thresholds to speed up the imputation analyses.

MINIMAC3 redefines the forward-backward equations in the reduced state space of the unique sequences. We have already introduced $L(.)$ as the forward equation for the original state space and we indicate the forward equation for the reduced state space at marker m as $\mathcal{L}(.)$. Consider a genomic segment bounded by markers p and q, and $k \in \{p, p+1, \ldots, q\}$. The forward equation is re-defined in the MINIMAC3 method in terms of reduced state space as follow:

$$\mathcal{L}(Z_p = u) = \sum_{n:\ h_n = y_u} L(Z'_p = n) \tag{1.21}$$

$$\mathcal{L}(Z_{k+1} = u) = \underbrace{\left((1-\rho_k)\mathcal{L}(Z_k = u) + \frac{N_u \rho_k}{U} \sum_{i=1}^{U} \mathcal{L}(Z_k = i)\right)}_{\text{transition}} \underbrace{Pr(t_{k+1}|Z_{k+1} = u)}_{\text{emission}} \tag{1.22}$$

$$L(Z'_q = n) = \mathcal{L}^{\mathrm{R}}(Z_q = u) \times \frac{1}{N_u} + \mathcal{L}^{\mathrm{NR}}(Z_q = u) \times \frac{L(Z'_p = n)}{\mathcal{L}(Z_p = u)} \tag{1.23}$$

Equation 1.21 folds probabilities at the start marker of the block. For all other markers in the block, forwards probabilities are computed using the reduced state space using Equation 1.22. Finally, probabilities are unfolded at the end of each block using Equation 1.23, returning to the original state space. In Equation 1.23, probability $L(.)$ is split into two parts, $\mathcal{L}^{\mathrm{NR}}(.)$ and $\mathcal{L}^{\mathrm{R}}(.)$, where $\mathcal{L}^{\mathrm{NR}}(.)$ denotes the left probability at marker q when no template switches occur between markers p and q; $\mathcal{L}^{\mathrm{R}}(.)$ denotes the probability when at least one switch occurs. At marker k, $\mathcal{L}^{\mathrm{NR}}(.)$ and $\mathcal{L}^{\mathrm{R}}(.)$ are defined as follows:

$$\mathcal{L}^{\mathrm{NR}}(Z_k = u) = \mathcal{L}(Z_p = u) \prod_{i=p}^{k-1} (1-\rho_i) Pr(t_{i+1}|Z_i = u) \tag{1.24}$$

$$\mathcal{L}^{\mathrm{R}}(Z_k = u) = \mathcal{L}(Z_k = u) - \mathcal{L}^{\mathrm{NR}}(Z_k = u) \tag{1.25}$$

By folding and unfolding reduced space probabilities at the beginning and end of each block, MINIMAC3 performs the forward-backward algorithm on a smaller state space and returns in the original state space only at the end of each block. A similar idea can be applied for the backward equation.

M3VCF file format M3VCF, which stands for "MINIMAC3 VCF", is a file format that can compactly store reference panel data. These files are created on the basis of the same idea as MINIMAC3. The idea is that storing the unique sequences of haplotypes over small segments is more convenient, as the expected number is less than the total number of haplotypes.

M3VCF files are formatted in a similar way as VCF files and therefore they are textual files that can be compressed using *deflate* algorithm. The first few lines are header lines and contain information on the number of blocks and segments. After the header, each segment is defined. In the model, segments must overlap at a common marker in order to allow folding and unfolding of the state space.

Imputation servers

In the last few years, imputation servers have been made available in order to provide free imputation and phasing service to the community. Users can upload phased or unphased GWAS genotypes and receive phased and imputed genomes in return. Imputation is typically performed using the 1000 Genomes (Phase 1 and 3) and HRC (McCarthy et al. 2016) reference panels.

There are two free imputation services currently available, the Michigan imputation server (Das et al. 2016)[8] and the Sanger imputation server (McCarthy et al. 2016)[9].

In particular, the Sanger imputation server performs imputation using an unpublished method based on the Positional Burrows-Wheeler Transform (PBWT) (Durbin 2014)[10]. The reference panel is stored using this data structure that allows very efficient compression due to LD. The imputation algorithm works by finding locally matching haplotypes to the target haplotypes. It then proceeds to impute performing a weighted sum of the alleles. This approach has the property that, once the haplotype data is placed into the PBWT data structure, the matching operations are independent of the number of haplotypes. The PBWT imputation step has

[8] `https://imputationserver.sph.umich.edu/index.html`
[9] `https://imputation.sanger.ac.uk/`
[10] A detailed description of the PBWT is given in Chapter 2.

complexity independent of the number of samples in the reference panel and does not rely on the Li and Stephens HMM. However, this also represents its limit, since imputation methods that use the Li and Stephens HMM are typically more accurate.

1.3.5 Imputation accuracy

In order to test accuracy of imputation methods, it is common practice to test imputation performance on a set of WGS data used as a gold standard. An experiment involves masking genotypes not in one of the commercially available SNP arrays and perform imputation using a dense reference panel, such as the HRC. Since we know the truth of the imputed sites, it is possible to compute the squared correlation between the true genotypes and the expected genotype derived from imputation. This expected value is called dosage. For target sample t at marker m, the dosage is defined as:

$$e_{tm} = \sum_{i=0}^{2} i \cdot p_{tmi} \tag{1.26}$$

where i represents the genotype and p_{tmi} is the genotype probability of sample t at marker m for genotype i derived from imputation. The true genotype is a discrete value between 0 and 2 and the dosage in a real number in the interval $[0, 2]$. Typically we compute the r^2 correlation in a single marker for each target sample, and we aggregate the r^2 into bins according to the minor allele frequency[11] of the SNP in the reference panel. In the n-th bin r-squared correlation between the imputed dosages and the true genotypes is calculated and plotted with the log allele frequency on the x-axis and the r^2 on the y-axis. An example of imputation accuracy plot can be found in Figure A.6.

[11]The x-axis can be defined as the minor allele frequency, the non-reference allele frequency or the minor allele count. For big reference panels the minor allele count provides a very intuitive way of visualising the data especially for big reference panels and rare variants. For this reason, we plot r^2 using MAC instead of MAF in Chapter 3

Reference panels

Imputation is based on identity-by-descent (IBD) and works well when there are long shared stretches between the study haplotypes and those in the reference panel. If an imputation reference panel contains several close relatives to the study samples, several real IBD segments can be copied from the reference panel, resulting in high accuracy.

For this reason, one of the most important factors that determines imputation quality is the number of haplotypes in the reference panel. The increased size of the reference panels allows to find longer stretches of matching sequence between study and reference haplotypes. In the last 10 years, reference panel evolved remarkably, increasing in sample size and ancestral diversity and even sequencing depth (Marchini 2019). Table 1.1 shows how reference panel size has increased over the years due to projects such as the International HapMap Project (The International HapMap Consortium 2007), the 1000 Genomes Project (1000GP) (The 1000 Genomes Project Consortium 2015), the UK10K Project (Huang et al. 2015), and the Haplotype Reference Consortium (HRC) (McCarthy et al. 2016). Even more, very soon larger reference panels will become available from the Trans-Omics for Precision Medicine (TOPMed) program (Brody et al. 2017; Taliun et al. 2019) and the 100,000 Genomes Project (Caulfield et al. 2019), both of which will exceed 50,000 high-coverage whole genome sequenced samples. Finally, sequencing data on all 500,000 participants of the UK Biobank (Bycroft et al. 2018) has recently been announced, and will soon be ubiquitously used as a reference panel, posing new computational challenges for imputation methods. A visualisation of the growth of reference panels is shown in Figure 1.1.

SNP arrays

SNP arrays are based on LD and are designed for different reasons, from capturing as much genetic variation as possible (aimed directly at GWAS) to increasing the performance of imputation (Bycroft et al. 2018). Some of them are more dense than others or specific for exon regions or explicitly contain known causal variants. For

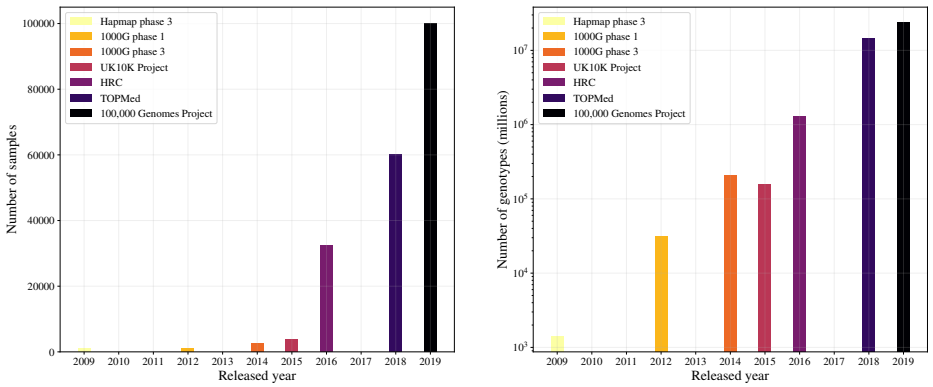

Figure 1.1: Size of major datasets used as reference panels in the last 10 years. Left: exponential growing in the number of samples for the major reference panels. The 100,000 Genomes Project and TOPMed are only estimates, as the datasets are not available yet. The figure on the right indicates that sequencing depth had also a similar trend. Here are shown the number of genotypes (number of markers time number of samples) of the same datasets. In order to be able to visualise smaller datasets, the y axis is on a log scale.

Reference panel	Yeas	Samples	Markers (Millions)
HapMap Project phase 3	2009	1,011	1.4
1000 Genomes phase 1	2012	1,092	29.0
1000 Genomes phase 3	2014	2,504	81.7
UK10K Project	2015	3,781	42.0
HRC	2016	32,470	40.4

Table 1.1: Evolution of imputation reference panels over time.

these reasons, imputation quality is affected by the choice of the SNP array (Marchini 2019). The relative performance of several SNP arrays is shown in Figure A.6.

Ancestry

The ancestry of the GWAS samples plays a role in imputation accuracy (Marchini 2019). It has been shown that imputation at common variance is more accurate in populations with higher LD, like Europeans, but at rare variants imputation is better for African populations. The hypothesis is that greater genetic diversity results in a larger number of haplotypes and this improves the chances that a rare variant is tagged by a characteristic haplotype.

Quality control

Before imputation, it is necessary to perform few steps of quality control (QC). Quality control pipelines subset both the number or markers and number of samples in a dataset, removing outliers and errors introduced during sequencing. As part of QC, it might be necessary to determine the number of valid variants, sample call rates, allele consistencies, strand flips, allele switches, minor allele frequencies, etc.

Quality control is a crucial step for imputation. Filters can reduce the number of variants of the study. For example, wrong genotype calls in a marker would affect imputation accuracy at the neighbouring markers. After the quality control step, pre-phasing is computed, typically one chromosome at the time, followed by imputation.

Chromosome X imputation

Chromosome X is a special case for imputation. Additional QC is required to check for the ploidy and imputation is typically split into two steps. The difference between chromosome X and other chromosomes is that chromosome X presents both pseudo-autosomal regions (PAR), where both male and females are diploids, and non-pseudo autosomal regions (non-PAR) where males are haploid and females are diploid. While imputation can be carried normally for PAR regions, non-PAR regions must be treated separately for males and females. Typically data is split into the PAR and non-PAR regions and the latter is split again by sex. Pre-phasing and imputation is then performed separately for non-PAR females and non-PAR males.

1.3.6 Power of association studies using imputed data

One way to test accuracy of imputation methods is to mask markers where the genotypes are known. Suppose to have a dataset of N samples and to have both true and imputed data for a certain marker. The imputed data has been obtained, for example, by masking the marker and running genotype imputation. As we covered in the previous sections, we can then compute the r^2 correlation of the dosage between the truth and the imputed data as a measure of imputation accuracy. It turns out that the r^2 is also a measure of the loss of power for

association testing when we use imputed data instead of real data. The power of using imputed data for the marker in question is r^2N (Pritchard and Przeworski 2001). Rephrasing, we can say that the effective sample size for association testing when using imputed data is reduced by r^2.

Possible sources of problems when using imputation data are described in the following two extreme examples. Suppose we have imputed data for GWAS where cases and controls for a binary phenotype have been genotyped using two different SNP micro arrays, and then imputed using the same reference panel. If a SNP has been poorly imputed in either cases or controls, this tends to have its allele frequency similar to the reference panel allele frequency (imputation suggests the genotype distribution of the reference panel at that marker) which can lead to false positive associations. Another problem can arise when a reference panel that is strongly genetically dissimilar with the GWAS dataset is used, leading to a potential increase of the noise in imputation estimates.

Another potential source of problem regards the effect size estimates. The effect size estimates from imputed data remain unbiased in a linear regression, provided the imputed variants are "well-calibrated" (Das 2017). Imputed genotypes are well-calibrated if the probability of observing a genotype at marker m is equal to the expected imputed probability. For example, if we consider a marker having an expected imputed probability of 0.5 for genotypes 0, "well-calibrated" means that a proportion close to 0.5 do indeed have genotype 0. The two extreme examples described above are two violations of the calibration assumption where the dosages do not represent correctly the alternate allele probability for GWAS samples[12].

Well-calibration

The well-calibration assumption is assumed to be true for GWAS using imputed data. However, there are several reasons that can cause violations of the assumption. For example, imputation relies on well-phased target samples and a problem in the phasing of the target samples can cause a violation of the well calibration

[12] A proof of asymptotic results for GWAS with imputed genotypes (using the well-calibration assumption) can be found in (Das 2017)), Appendix B.

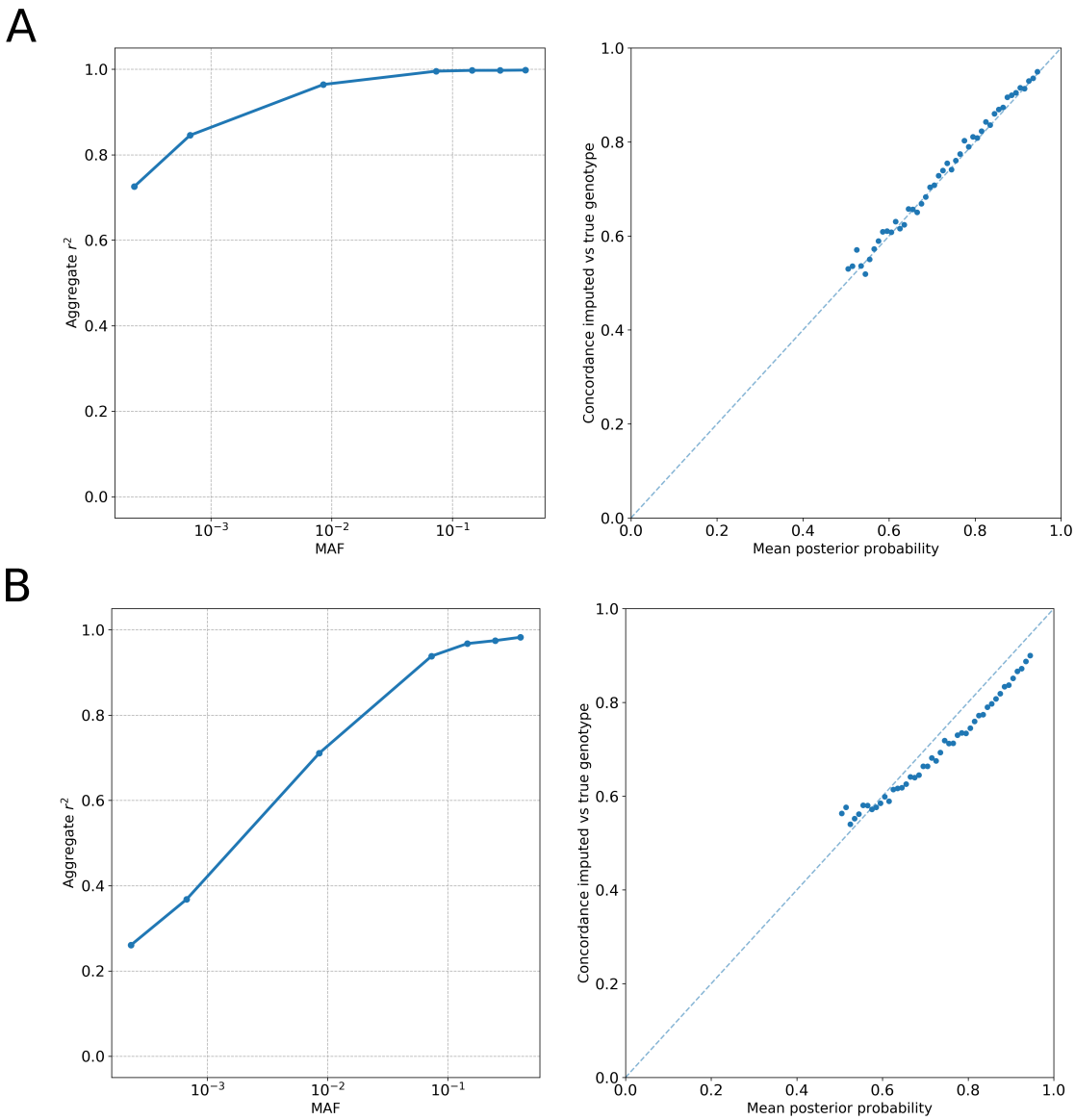

Figure 1.2: Imputation accuracy and calibration of posterior probabilities for the whole 10Mb imputed region. To evaluate calibration, we split imputed genotypes into bins according to their posterior probability distribution. We plot the mean posterior probability in each bin (x-axis) against the percentage of correctly predicted genotypes in each bin (y-axis). The perfectly phased dataset is shown in part (A) and the dataset containing phasing errors is shown in part (B).

assumption. This is especially true for methods like BEAGLE5 and IMPUTE5 that subset the state space.

In order to show how bad phasing can cause violations from the well calibration assumption, we used a simulated dataset of 1,000 samples composed of 3,333 markers

and 223,116 masked variants on a 10Mb region of perfectly phased data[13]. We then re-phased the dataset by running SHAPEIT4 with minimum settings, introducing approximately 6% switch error rate. Finally, we imputed both the perfectly phased dataset and the dataset with phasing errors using BEAGLE5 with simulated 10,000 samples reference panel from the same region.

To assess the calibration of imputation probabilities, we computed the concordance rate between the most likely genotype and the true genotype. For well-calibrated probabilities, we expect that predicted genotypes with posterior probability α will have concordance rate of approximately α.

Figure 1.2 (left) shows the r^2 correlation between imputed and true genotypes for the perfectly phased dataset (part A) and the dataset having a 6% switch error rate (part B). The effect of phasing errors affect mainly imputed genotypes at the rare frequency spectrum. Figure 1.2 (right) shows the calibration of the posterior probabilities. Phasing errors affect the calibration and the figure indicates that the imputation method is overconfident in the case of phasing errors.

Info-score

When researchers perform GWAS and run imputation on a study dataset, a natural question that rises is how good is the quality of imputation at each marker, since the true value for variants that are not genotyped is not available, in order to discard badly imputed sites. For this reason, several imputation quality scores have been introduced, giving a score value of the imputation quality at each marker. The IMPUTE model makes use of the so-called info-score. Assuming imputed genotypes are well-calibrated, the info-score in a marker defines a measure of the difference of information between imputed markers (with an imputation method) and the information obtained if only the allele frequency is used to impute the marker (Marchini and B. Howie 2010).

Suppose we have imputed data for a set of N individuals. We indicate as $p_{imk} = Pr(g_{im} = k|H, T)$ the probability, obtained from imputation, that the

[13]Details of how we generated the simulated dataset and the reference panel can be found in Section 3.3.

genotype at the mth SNP of the ith individuals is k, where $k \in \{0, 1, 2\}$, and we indicate the imputed genotype distribution for the individual i at marker m as $\boldsymbol{p}_{im}$. The expected allele dosage for the ith individual at genotype at the mth SNP is $e_{im} = p_{im1} + 2p_{im2}$. The variance of the imputed genotype distribution at marker m for individual i is denoted as v_{im}, where:

$$v_{im} = \mathbb{E}(e_{im}^2|\boldsymbol{p}_{im}) - \mathbb{E}(e_{im}|\boldsymbol{p}_{im})^2 = p_{im1} + 4p_{im2} - e_{im}^2 \tag{1.27}$$

Let us consider the two extreme cases. First, when there is no uncertainty in the distribution of $\boldsymbol{p}_{im}$, the value of v_{im} is zero. Second, if imputation does not provide any additional information over what we had by only knowing the allele frequency at marker m, then the variance of the imputed genotype corresponds Hardy-Weinberg variance $w_{im} = 2f_m(1 - f_m)$ where f_m is the allele frequency of the allele 1. This can be interpreted as the fact that imputation has not provided any new information compared to what was already available, based on the allele frequency f_m. It follows that the ratio of the variances v_{im}/w_{im} is of particular interest to evaluate the information of imputed data over information we have only using HWE. We can then define the imputation info-score metric for locus m as:

$$INFO_m = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{v_{im}}{w_{im}} \tag{1.28}$$

The info-score is bounded above at one and equals zero when the mean variance of imputed genotypes equals the HQ variance. This measure is used by several versions of the IMPUTE software and it is commonly used as a filter to discard imputed SNPs in GWAS datasets, e.g. discarding all the variants with info-score < 0.3.

1.4 The haplotype phasing problem

Haplotype estimation, or phasing, is a statistical process of inference of haplotype information from genotype or sequenced data. Phasing is used as a necessary step in several areas of genetics, including genome-wide association studies (Balding

2006), fine-mapping of complex disease genes (Tewhey et al. 2011) and population studies (Tishkoff et al. 1996).

In this section, we define the phasing problem following the lines of Gusfield 2004. Let G be a set genotype vectors $g_i = \{g_{i1}, \ldots, g_{iM}\}$ defined on M markers, having values in $\{0, 1, 2\}$, where 1 indicates a heterozygous site. A solution to the phasing problem is a set of K pairs of binary vectors $h_i^{(1)}, h_i^{(2)}$, called diplotypes, one pair for each genotype vector, where $h_i^{(1)} = \{h_{i1}^{(1)}, \ldots, h_{iM}^{(1)}\}$ and $h_i^{(2)} = \{h_{i1}^{(2)}, \ldots, h_{iM}^{(2)}\}$. Both of the binary vectors of the haplotype pair $\{h_i^{(1)}, h_i^{(2)}\}$, associated with a genotype vector g_i, have fixed value 0 or 1 in any homozygous position where g_i has value 0 or 2: therefore, the phasing problem is trivial for homozygous sites. However, any position where g_i is heterozygous and has value 1, only one of $h_i^{(1)}, h_i^{(2)}$ must have value 0, while the other has value 1.

A diplotype is compatible with a genotype g_i if the sum of their alleles in every position is equal to g_i, $g_i = h_i^{(1)} + h_i^{(2)}$. A site m in the genotype vector g_i is called ambiguous if its value is 1, and a genotype vector is called ambiguous if it contains at least two ambiguous sites. For an individual with k ambiguous sites, there are 2^{k-1} consistent distinct haplotype pairs. This is the main reason that makes the phasing problem difficult.

Phasing methods such as SHAPEIT4 or EAGLE2 (Delaneau et al. 2018; Loh et al. 2016), can use a reference panel of haplotypes to help the phasing process and improve accuracy by taking information from already phased, accurate haplotypes. Phasing is usually an iterative approach (e.g. in the form of a Gibbs sampler), where at each iteration, new estimates of each sample's haplotypes are sampled using the current information about haplotypes of all other individuals. An iteration is typically a visit of all the sampled haplotypes in a random order, and the process stops after few iterations, when some convergence criteria is met.

1.4.1 Phasing models

EAGLE2

EAGLE2 (Loh et al. 2016) is a phasing method that uses information from an external reference panel of haplotypes. The linear complexity of this approach is realised through an efficient data structure representing the reference panel, combined to a rapid search algorithm that explores only the most relevant paths through a model similar to the diploid Li-Stephens HMM. The data structure used for the reference panel, called HapHedge-Multi, is built from the Positional Burrows-Wheeler Transform (see Chapter 2 for details).

EAGLE2 does not rely on the standard forward-backward HMM recursions, but introduces non-Markov recombination probabilities that model the coalescent process more accurately, following the idea that moving along the genome, the probability of recombination should decrease as the sharing length grows.

The use of the HapHedge-Multi data structure is the key feature of the method. HapHedge-Multi compresses multiple haplotype segments of the reference panel to their subset of unique sequences, allowing fast look-up haplotype frequencies in arbitrary segments. The Positional Burrows-Wheeler Transform is used as a fast (linear) method to generate trees of sequence segments, by exploiting the relationship of the Positional Burrows-Wheeler Transform with (reverse) prefix trees.

SHAPEIT4

SHAPEIT4 (Delaneau et al. 2018) is the latest version of the SHAPEIT method. In 2008, SHAPEIT1 was introduced. The main advance of SHAPEIT1, compared to other methods at the time, was the linear complexity of the HMM calculations. This was achieved by using a graphical model to represent all of the possible haplotypes underlying a given individual's genotypes.

The key idea is that the distribution of paths through the graph conditional upon a set of known haplotypes can be modelled as a Markov Chain. The transition probabilities of this Markov Chain can be learned using the forward-backward algorithm for HMMs. Pairs of haplotypes consistent with the genotype can then be

sampled, and usually this is followed by a pruning step, removing states that seem unlikely given the data and then merging chunks together. This approach has the property that the model gradually reduces the solution space as the method proceeds.

SHAPEIT2 is a widely used phasing method that chooses a local subset of conditioning haplotypes using Hamming distance (as in IMPUTE2). In practice, Hamming distance calculations require a quadratic scan in the number of haplotypes, which becomes prohibitive when facing large sample sizes.

SHAPEIT4 performs the copying state selection using the Positional Burrows-Wheeler Transform (see Chapter 2 for details), reducing the selection part of the algorithm to linear complexity. In addition to this, states selected using the PBWT are also very precise, and the model has the feature which automatically increases the number of states retrieved if the current haplotype is not well-represented the in the set of states. More details about the SHAPEIT4 model can be found in the IMPUTE5 description in Figure 3, since the models share similarities in the copying state selection algorithm.

1.5 Summary and discussion

In this chapter, we briefly introduced genome-wide association studies in order to present the genotype imputation problem. We described the statistical framework of genotype imputation and outlined the current state-of-the-art imputation algorithms. We saw how current genotype imputation is typically performed on imputation servers and makes use of highly-efficient data structures in order to scale efficiently with the size of current datasets. We also introduced the haplotype phasing problem, which has very wide applications and shares models with imputation.

The rest of the dissertation is structured as follows. Chapter 2 shows an efficient representation of the Positional Burrows-Wheeler Transform that can be used as a library for other researchers. In Chapter 3, we introduce a new method for genotype imputation and compare it with the leading methods in the field. In Chapter 4, we evaluate the copying state selection algorithms used in literature. In Chapter 5, we introduce a simple metric to measure the expected amount of

sequence copied between different haplotypes and populations in 1000 Genomes Project. In Chapter 6, we introduce a pipeline that performs quality control and genotype imputation on both a GWAS study dataset and a reference panel of haplotypes, involving pre-phasing. Chapter 7 shows an important application of our imputation method on the two major British genetics datasets: 100,000 Genomes Project and UK Biobank. Imputation of the UK Biobank dataset, using 100,000 Genomes Project as a reference panel, involves careful quality control, software engineering techniques and accurate post-imputation data analysis in order to give to the scientific community a new and accurate resource that can potentially undercover new discoveries in complex disease genetics.

2

An Introduction to the Positional Burrows-Wheeler Transform

Contents

2.1 Introduction

The Positional Burrows-Wheeler Transform (PBWT) (Durbin 2014) is a data structure that represents a set of haplotypes using the reverse-prefix sort order. The PBWT is a special case of the Burrows-Wheeler transform (Burrows and Wheeler 1994), which is a method for string data compression defined on a limited alphabet and the FM-index (Ferragina and Manzini 2000), which is a data structure that allows to locate the position of each occurrence of a pattern in a Burrows-Wheeler transform of strings with sub-linear complexity with respect to the size

of the input data.

The PBWT is connected to the suffix array theory, but, in this context, reverse prefixes are considered rather than suffixes, reversing the order of the standard string matching theory. The reason for this reverse order is that several algorithms and data structures in genetics are defined and follow the forward order from marker 1 to M, rather than backwards.

The main advantage of the PBWT and all its complementary data structures, is that it allows fast string matching. For example, sub-string matching of a new sequence (haplotype) in a set of haplotypes represented as a PBWT has time complexity independent on the number of haplotypes in the PBWT set and is only linear in the number of markers, $O(M)$. This property makes PBWT a particularly efficient data structure that allows fast querying: implementations, such as BGT (H. Li 2016), can be used to store and query the haplotypes of thousands of samples using the PBWT.

The data structure has been successfully used for haplotype phasing (Loh et al. 2016; Delaneau et al. 2018) and genotype imputation[14]. Our imputation method, IMPUTE5, uses the data structure to select a set of highly-confident states to run a sub-linear HMM for imputation. Details can be found in Chapter 3.

In this section, we give an overview of the data structure, providing details of its construction and the main string matching algorithms. We show the sub-linear string matching behaviour and dramatic compression rates it reaches on real data using our PBWT implementation, called PBWTcpp. PBWTcpp is publicly available at: `https://github.com/SimoneRubinacci/PBWTcpp`.

2.1.1 PBWT arrays

The PBWT itself is only a transformation (function) of the original set of haplotypes, obtained by mapping binary haplotype data in a specific sort order. However, when we refer generically to the PBWT, we often refer to a set of data

[14] An unpublished genotype imputation method is available on the Sanger imputation server.

structures, such as the positional prefix array or the divergence array, other than the PBWT itself. We will define these data structures in the next sections.

The notation used here is different from the original notation used by Durbin (Durbin 2014); we adopt a notation similar to the one used by Li et al. (H. Li 2016).

2.1.2 Notation and definitions

The PBWT is a way to encode binary matrices, which is especially useful in the case of haplotypes defined on a set of binary markers, each with two alleles arbitrarily coded as 0 and 1. Let $H = \{h_1, h_2, \ldots, h_N\}$ be a set of N haplotypes genotyped at M markers, $h_n = \{h_{n,1}, h_{n,2}, \ldots, h_{n,M}\}$, where $h_{n,m} \in \{0, 1\}$, represents the nth haplotype value. It is useful to think of H as a $N \times M$ binary matrix, where each entries are defined by the reference haplotype (row) and marker (column). For this reason, we refer to H using the usual matrix notation (one-indexed instead of zero-indexed).

The PBWT of H, indicated as Y, is another $N \times M$ binary matrix, where the mth column of Y is an invertible transformation of the mth column of H. The PBWT Y is complemented by another $N \times (M + 1)$ matrix A, where every column $A_{:,m}$, called the *positional prefix array*, is a permutation of $\{1, 2, \ldots, N\}$ which defines the reverse-prefix sort order of the haplotypes in H up to marker m.

We use the word reverse-prefix to indicate a suffix read in the reverse order, from the right to the left. Formally, define the binary string $h^r_{n,1:m}$, where the superscript r stands for reverse, as the reverse-prefix of the nth haplotype ending at marker m:

$$h^r_{n,1:m} = h_{n,m} h_{n,m-1}, \ldots, h_{n,1} \tag{2.1}$$

and let $\{h^r_{n,1:m}\}_n$ be the set of all the N reverse prefixes at marker m. We can now define $A_{n,m}$ to be the index of the nth lexicographically sorted reverse-prefix along the set $\{h^r_{n,1:m}\}_n$.

It follows from the definition that $A_{:,m}$ represents a bijection on $\{1, \ldots, N\}$ and thus is invertible. As a special case, overloading the standard matrix notation,

we define $A_{n,0} = n$, representing the order of empty reverse prefixes as the identity permutation.

The PBWT Y is directly derivable from H and the prefix array A:

$$Y_{n,m} = H_{A_{n,m-1},m} \tag{2.2}$$

in other words, the PBWT at marker m is the vector of values of the haplotypes in H at marker m, in the order defined by the positional prefix array at marker $m-1$.

One use of $Y_{:,m}$ is to update $A_{:,m-1}$ to $A_{:,m}$. Suppose b is a symbol, $b \in \{0, 1\}$, we can define a mapping between positional prefix array at markers $m-1$ and m:

$$\phi_m(n) = c_m(Y_{:,m}) + rank_m(Y_{n,m}, n) \tag{2.3}$$

where $c_m(b)$ gives the number of symbols in $Y_{:,m}$ that are lexicographically smaller than b and $rank_m(b, n)$ the number of b symbols in $Y_{:,m}$ before position n. The nth haplotype in the positional prefix order at column $m-1$ is ranked $\phi_m(n)$ in column m. Thus

$$A_{\phi_m(n),m} = A_{n,m-1} \tag{2.4}$$

Equations 2.2 and 2.4 give a procedural algorithm to compute $Y_{:,m}$ and $A_{:,m}$ from $H_{:,m}$ and $A_{:,m-1}$. Figure 2.1 shows an example to visualise how values of the column $Y_{:,m}$ are determined using the sort order of prefixes up to position $(m-1)$.

When representing haplotype data, Y is also strongly compressible. This emerges from the fact that there is strong correlation between adjacent markers in H due to linkage disequilibrium, and therefore there are long runs of the same symbol in the columns of Y. This makes columns of Y much more compressible than the columns of H.

Pseudo code to build the positional prefix array $A_{:,m}$ from $A_{:,m-1}$ is given in Algorithm 1. We can therefore calculate the entire set of orderings for all markers in a single pass through all the sequences, in time $O(NM)$.

The cost of building a PBWT Y from H is $O(NM)$, including all the complementary matrices. Using the PBWT indices, it is possible the set of find maximal

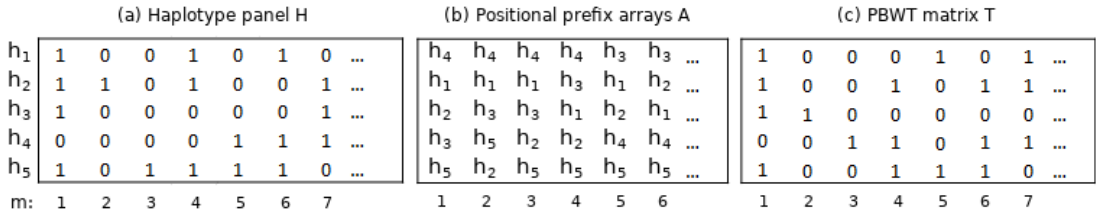

Figure 2.1: PBWT example. A visualisation of the the original set of haplotypes H (a), the positional prefix array A (b), and the PBWT matrix Y (c). A region with $M = 7$ markers and $N = 5$ haplotypes $\{h_1, h_2, ..., h_5\}$ is considered. The special case $A_{:,0}$ is not visualised. As expected, the columns $Y_{:,0}$ and $H_{:,0}$ have the same values, because $Y_{:,0}$ has been built using the identity permutation $A_{:,0}$. h_4 is at the top of the column $A_{:,1}$ because h_4 is the only haplotype that has value 0 at marker 1, and all the other haplotypes preserve the previous (identity) order. Using $A_{:,1}$ and $H_{:,2}$ Algorithm 1 builds the column $Y_{:,2}$ which is a transformation of $H_{:,2}$. For example for $n = 3$, $Y_{3,2} = H_{A_{3,1},2} = H_{2,2} = 1$.

Algorithm 1 Builds the positional prefix array $A_{:,m}$ from $A_{:,m-1}$ for $m \in \{1, \ldots, M\}$

```
u ← 0
v ← 0
if m == 1 then
    for n ← 1 to N do
        A_{n,0} ← n
    end for
end if
Create empty arrays a[], b[] of size N
for n ← 1 to N do
    if H_{n,m} == 0 then
        a[u] ← A_{n,m-1}
        u ← u + 1
    else
        b[v] ← A_{n,m-1}
        v ← v + 1
    end if
end for
A_{:,m} ← concatenation of a[1, ..., u] followed by b[1, ..., v]
```

matchings within H in linear time and the set of maximal matchings of a new sequence z in H in $O(M)$, independently from the number of haplotypes in H.

Divergence arrays and FM-index

Y and A represent only the basic form of the PBWT. It is possible to complement them by storing additional information such as the rank indices U, V (usually called

FM index) and the divergence array. Each column m of these two matrices stores information about the $rank_m(a, m)$ for symbol $a = 0$ and $a = 1$ respectively. At marker m, there arrays represent sequences of increasing numbers, starting from zero and having the property that $U_{n,m} + V_{n,m} = i$, for $i \in \{1, 2, \ldots, N\}$. We also need to introduce another vector called c, representing the number of 0 symbols at each maker[15]. Using the rank arrays and the vector c, it is possible to locate reverse-prefixes into PBWT by only using $O(1)$ operations at each marker.

Another important data structure is the divergence matrix D. Columns of D contain the position of the last reverse-prefix mismatch between adjacent haplotypes in the order A. The value of $D_{n,m}$ is defined to be the smallest value m' where the sequence $h^r_{n,1:m}$ matches $h^r_{n-1,1:m}$. In the case of a mismatch, the value of $D_{n,m}$ is set to m. We note that the start of any maximal match ending at m between any $\{h^r_{i,1:m},\ h^r_{j,1:m}\}, (i < j)$ is given by:

$$max_{i<n\leq j} D_{n,m} \tag{2.5}$$

By exploiting Equation 2.5, we can efficiently extend Algorithm 1 to update D as we sweep through the data, as shown in Algorithm 2.

2.1.3 String matching algorithms

The most important feature of the PBWT is the efficiency when it is used for string matching. String matching on a set of haplotypes is a common task in the field of human genetics and it is used, between others, by IBD detection algorithms, haplotype phasing and genotype imputation.

Few definitions about string matching are needed. Suppose H to be a set of N haplotypes defined on M markers. We say that there is a match between two haplotype sequences x, y, from marker m_1 to marker m_2, where $(m_1 < m_2)$, if there is a region $[m_1, m_2)$ where the haplotype data for x and y from marker m_1 (included)

[15]In the typical FM-index setting, a more general C table is used, in which for every symbol a in the alphabet $C_{a,m}$ returns the number of occurrences of characters lexically smaller than a up to marker m. However, since we are using a binary alphabet, there is no need to define $C_{1,:}$. For this reason, $C_{0,:}$ can be reduced to c as defined in the main text, representing the number of occurrences of zeros at the marker.

Algorithm 2 Builds the positional prefix array $A_{:,m}$ from $A_{:,m-1}$ and the divergence array $D_{:,m}$ from $D_{:,m-1}$ for $m \in \{1, \ldots, M\}$

```
1: u ← 0
2: v ← 0
3: p ← m
4: q ← m
5: if m == 1 then
6:     for n ← 1 to N do
7:         A_{n,0} ← n
8:         D_{n,0} ← 0
9:     end for
10: end if
11: Create empty arrays a[], b[], d[], e[]
12: for i ← 1 to N do
13:     if D_{i,m} > p then
14:         p ← D_{i,m-1}
15:     end if
16:     if D_{i,m} > q then
17:         q ← D_{i,m-1}
18:     end if
19:     if Y_{i,m} == 0 then
20:         a[u] ← A_{i,m-1}
21:         d[u] ← p
22:         u ← u + 1
23:         p ← 0
24:     else
25:         b[v] ← A_{i,m-1}
26:         e[v] ← q
27:         v ← v + 1
28:         q ← 0
29:     end if
30: end for
31: A_{:,m} ← concatenation of a[1, ..., u] followed by b[1, ..., v]
32: D_{:,m} ← concatenation of d[1, ..., u] followed by e[1, ..., v]
```

to m_2 (excluded) matches. We indicate a match as $x[m_1, m2) = y[m_1, m_2)$. We say that this match is locally maximum if it cannot be extended in any of the directions (by decreasing m_1 or increasing m_2). When comparing a sequence x to a set of sequences H, we say that x has a set-maximal match between m_1 and m_2 with a sequence $h_n \in H$ if the match is locally maximum and there are no other matches from x to any other sequence in H for any extension of interval $[m_1, m_2)$. In other

words, the set of maximal matches between a sequence x and a set of haplotypes H that includes the marker m is then the set of haplotypes in H that shares the longest match with x in a region that includes marker m.

Haplotypes that share a reverse-prefix cluster next to each other in the PBWT, forming a 'block'. In other words, blocks define distinct haplotypes, considering reverse-prefixes of length l, where l can be arbitrary chosen. At marker m, a block is defined as the maximal interval of indices i and j, $i < j$, where for each $n \in (i, j]$, the property $D_{n,m} <= m - l$ holds.

The PBWT provides three main algorithms for string matching:

- finding matches between sequences in H of length longer that a parameter l. The algorithm works by scanning the blocks of haplotypes in the divergence arrays. This scan is linear in N. The overall cost of the algorithm is $O(\max\{NM, \#\text{ matches}\})$

- finding all the set-maximal matches between sequences in H. Like before, the algorithm works by examining block information in the divergence array reporting a match only when it cannot be longer extended. The cost is $O(NM)$, under the condition that there are not arbitrarily large groups of identical sequences from 0 to N.

- finding all the set-maximal matches from a new sequence x to H. The first step of the algorithm works by searching x in H using the rank operations. During this search, block information of x in H is kept, reporting the set-maximal match when the block ends. It is then necessary to search back at the end of a matching, in order to find the longest match (excluded the set-maximal) until that marker. The algorithm runs in $O(NM)$, even if a back search is performed. A variant of this algorithm that performs only the search step is used by IMPUTE5 selection algorithm (see chapter 3).

2.2 Methods

We developed a software package for the PBWT, called PBWTcpp. The package is an independent implementation based on Richard Durbin's PBWT package[16]. The software has been developed in C++ and aimed to be used as a library inside other packages, by maintaining a minimal set of classes that perform basic operations on the PBWT data structure.

PBWTcpp has the following features:

- can read a set of haplotypes using VCF/BCF file formats.
- represents a set of haplotypes in a PBWT;
- can read and store the PBWT using the compressed PBW3 file format;
- implements string matching algorithms;
- can compute the reverse PBWT from a forward PBWT.

PBWTcpp splits information by chromosome and requires to specify a parameter indicating a region of the chromosome. The use of the index allows the program to quickly locate the start and the end location of the region of interest in the file. A PBWT of the data is then created and stored compressed in memory.

The generated PBWT can be stored in a file using the PBW3 format[17]. A PBWT3 format can also be used as an input: in that case Y is read from memory and there is no need to compute Y again.

The positional prefix arrays and the divergence array are not stored in memory, but they are updated at every marker. It is possible to exploit the PBW3 run-length encoding representation to read blocks of haplotypes and update the positional prefix arrays in blocks as well. This allows very fast reading time from the run-length encoded dataset.

PBWTcpp is publicly available at: `https://github.com/SimoneRubinacci/PBWTcpp`.

[16] `https://github.com/richarddurbin/pbwt`

[17] Proposed by Durbin 2014, it is a binary file format that represents the PBWT using run-length encoding compression on the columns of Y.

2.2.1 Data

We used phased chromosome 10 data from three large human datasets to test PBWTcpp: the 1000 Genomes Project (The 1000 Genomes Project Consortium 2015), the Haplotype Reference Consortium (McCarthy et al. 2016) and the UK Biobank (Bycroft et al. 2018). Details of these datasets will be explained in the following chapters. Summary of the main statistics of the data used is shown in Table 2.1.

Datasets	**Phased**	**Number of Samples**	**Number of Markers**
1000GP chr 10	Yes	2,504	2,335,930
HRC chr 10	Yes	32,470	1,927,503
UKB chr 10	Yes	487,442	32,626

Table 2.1: Sample size and number of markers of the datasets used.

The datasets have a very different number of samples and markers. For example, the 1000GP contains approximately 200 times less samples than the UK Biobank, whereas it contains 75 times more markers. We used these datasets to test the performance of the PBWT in these two extreme cases, in terms of compression rates and speed. The HRC is the biggest dataset of the three considered here, and represents the current state-of-the-art reference panel for imputation and phasing.

2.3 Results

Compression

One of the properties of the PBWT is that adjacent columns of Y are correlated due to linkage disequilibrium. Typically, only Y is stored in memory, and columns of the positional prefix array and the divergence array are updated at every marker. The PBW3 file format is composed of a file with extension *.pbwt*, containing the compressed matrix Y and few auxiliary files containing other information like sites and samples. These files are usually stored as plain text, but could potentially be

Dataset	vcf	vcf.gz	PBW3 (.pbwt)	PBW3 (all)
1000GP chr 10	23GB	612MB	56MB	94MB
HRC chr 10	260GB	3.8GB	192MB	222MB
UKB chr 10	60GB	3.1GB	443MB	450MB

Table 2.2: Memory used to store haplotype data for chromosome 10. Uncompressed VCF, gzipped VCF and PBW3 file formats are considered. For the PBW3 file format both the binary (.pbwt) and the sum of the binary file plus all the auxiliary files are shown.

compressed using gzip. A comparison of the size of uncompressed, gzipped and PBW3-compressed files is shown in Table 2.2.

On real data from chromosome 20, the PBWT seems to be able to compress the dataset from 10 to 20 times more than the standard gzipped VCF file format. The UK Biobank dataset has the lowest PBW3 compression rate compared to the gzipped VCF. The reason for this is that the UK Biobank dataset contains SNP array data and therefore is defined on a sparse set of markers. The two other datasets use other sequencing technologies and, effectively, have more markers. In this case, there is more LD between adjacent markers and thus a bigger compression rate is obtained by the PBW3 format. Compression also depends on the different allele frequency spectrum. For these reasons, the PBWT can be a very efficient method to store WGS datasets, especially with big sample sizes.

Basic operations

We measured the computational time of four basic operations: (i) build a PBWT from a VCF file containing a set of haplotypes, (ii) build a reverse PBWT from a PBWT stored in memory, (iii) scan all the markers of a PBWT updating the positional prefix array and the rank arrays, (iv) scan all the markers of a PBWT updating all the arrays (same as point (iii) but including the divergence array).

Operations (i) and (ii) represent two common operations in the PBWT. Usually we start from a set of haplotypes stored in a VCF file and build a PBWT representation in order to perform string matching.

Dataset	Build PBWT	Build reverse	Read (ranks)	Read (div)
1000GP chr 10	10:02	00:55	00:03	00:24
HRC chr 10	59:14	11:20	00:38	04:20
UKB chr 10	16:56	04:10	00:19	01:24

Table 2.3: Time (mm:ss) spent to build a PBWT, the reverse PBWT and reading time for different computed PBWT arrays

Operations (iii) and (iv) relate to string matching algorithms, used to find matchings within the PBWT sequences or between the sequences stored as a PBWT and external set of haplotypes. This latter case is the core of the state selection algorithms used for imputation and phasing presented in chapter 3.

Table 2.3 shows the time required to compute operations (i-iv) for the considered datasets. The additional time spent to compute the divergence array divergence array (iv) compared to time used when only rank arrays are computed (iii) is of particular interest. For this reason, two different algorithms for state selection will be proposed in chapter 3: a faster, approximate version that does not require the calculation of the divergence array, and a slower but more accurate state selection metric that makes use of the information contained in the divergence array.

Matching algorithms

PBWTcpp implements the three matching algorithms presented in the previous section:

1. finding matches between sequences in the PBWT of length longer that a parameter l indicating the number of markers.

2. finding set-maximal matches between sequences in the PBWT.

3. finding all the set-maximal matches from a new sequence to the sequences in the PBWT.

For the last of these algorithms, PBWTcpp requires an external set of sequences in a VCF format and finds all the set-maximal matchings between each of the

Dataset	Matchings	Max length	Time
1000GP chr 10	2	21927	00:51
HRC chr 10	4	110927	07:24
UKB chr 10	1	31610	02:25

Table 2.4: Longest matches within the test datasets and time required by the algorithm.

external sequences and the PBWT. Several algorithms in genomics perform this operation routinely, such as haplotype phasing or IBD detection methods.

For each of the testing datasets, we found the longest matches by choosing appropriate values of the parameter l of the matching algorithm 1. For the 1000GP dataset, we used l=20,000. For the HRC reference panel, since the amount of samples is more than two times bigger, we needed to increase the parameter to l =100,000. Finally, for the UK Biobank that represents SNP array data, we used the value l =30,000. The longest matches within the datasets are shown in Table 2.4.

2.4 Summary and discussion

In this section, we focused on the Positional Burrows-Wheeler Transform data structure to store a set of haplotypes. We explained the advantages of representing haplotypes in the reverse prefix order, showed the linear to sub-linear time complexity of the string matching algorithms and the strong compression it achieves by exploiting linkage disequilibrium.

The PBWT has also limitations. Ideally, if the full PBWT and all the data structures are available in memory, several algorithms based on string matching would have purely sub-linear computational cost. However, this is not the case, as a full representation of the PBWT data structures in memory is too expensive and compression algorithms are required, increasing the cost to linear.

Good compression is possible for the columns of Y but also the rank arrays could be compressed fairly well using differential encoding, since we expect long LD blocks in the columns of the PBWT. However, the main problem is to compress

the positional prefix array, since it stores permutations of integers. It has been proposed to store a positional prefix array at sparse positions, e.g. every 32nd or 64th marker, but requires some computation after the look-up. A common solution is then to build the PBWT on-the-fly and store one column of all the auxiliary data structures at the time, by only doing a single pass of the data.

In the next chapter, we present a genotype imputation method that makes use of a PBWT representation of the reference panel. It exploits the PBWT representation in order to select a subset of highly accurate "matching" haplotypes and obtain sub-linear running time for the subsequent and most expensive parts of the algorithm. More and more methods in the field are now using the PBWT and it appears that the use of suffix arrays to represent haplotype data will be dominant in the future.

3

A Genotype Imputation Model for next-generation datasets

Contents

3.1 Introduction

Imputation is usually performed as part of a GWAS, where individuals are genotyped using SNP array technology, having between 300,000 to 5 million markers

across the genome, using a high coverage sequenced reference panel of haplotypes with tens of millions of markers (Marchini and B. Howie 2010). For example, a recent study imputed the UK Biobank dataset (Bycroft et al. 2018) by increasing the number of testable markers from 825,927 to over 96 million. Details of the genotype imputation problem can be found in chapter 1.

In this chapter we present IMPUTE5, a haploid genotype imputation method designed for the new generation of reference panels. The main features of the method are: the use of new indexed reference panel file format, the PBWT to select a subset of reference panel haplotypes in order to reduce the state space in the IMPUTE model, and delayed imputation directly into BGEN file format, specifically designed for imputation data.

We benchmark our imputation method against other methods using simulated reference panels up to 1,000,000 haplotypes in size, and real reference panels such as the 1000 Genomes Project (The 1000 Genomes Project Consortium 2015) and the Haplotype Reference Consortium (McCarthy et al. 2016).

3.2 Methods

IMPUTE5 is based on the haploid IMPUTE model. It requires pre-phasing and for each target haplotype computes the forward-backwards probabilities of haploid Li and Stephens hidden Markov model (N. Li and Stephens 2003). It finally performs imputation using linear interpolation from the two neighbouring states of the model.

The computational performance of the model is made possible by adopting previous advances with several improvements. In the next few sections we provide a detailed description of the model.

Emission probability

IMPUTE5 models the emission probability of Equation 1.13 differently to the standard IMPUTE model, adopting the simpler version proposed by the SHAPEIT4

model (Delaneau et al. 2018), reducing the equation to:

$$Pr(t_m = a|Z_m = n) = \begin{cases} 0.9999 & \text{if } h_{n,m} = a \\ 0.0001 & \text{otherwise.} \end{cases} \tag{3.1}$$

where $t = \{t_1, t_2, \ldots, t_T\}$ is a target haplotype and t_m is the value at marker m, Z_m is the haplotype copying label at marker m, and $a \in \{0, 1\}$ is a haplotype value. It has been shown that imputation is relatively insensible to the mutation parameter (B. L. Browning and S. R. Browning 2016) and we verified that the adoption of this emission probability increases slightly accuracy especially for big reference panels.

3.2.1 PBWT for state selection

One of the key features of IMPUTE5 is the state selection algorithm. The method represents the reference panel at target markers using the PBWT and exploits the fast search properties of the data structure in order to identify a subset of states that share long stretches with the target haplotypes. Haplotypes that share long stretches are usually called to be in identity-by-state (IBS). This subset of states is then used to defined the state space of the Li and Stephens HMM calculations. By selecting a small set of IBS states, the computation of the forward-backwards probabilities of the HMM scales sub-linearly with the size of the haplotypes in the reference panel. In the IMPUTE5 model we assist to a decrease in the number of selected states when the number of reference haplotypes grows, due to larger IBS segments.

The PBWT of the reference panel at genotype sites is built at run-time across the region being imputed, one marker at the time. The selection algorithm is performed at the same time, by using the rank arrays at the marker. This means that, after one pass of the full dataset, the state selection has been performed for *all* the target haplotypes. Therefore, there is no need to store the full PBWT of the reference panel in memory, but positional prefix arrays and ranks arrays are constantly updated.

At marker m, the selection algorithm works by locating the longest reverse-prefix shared between the target haplotype and the reference panel up until marker m, and selecting a set of reference haplotypes "close" to the target haplotype to add to the list of copying states. Intuitively, since the PBWT contains lots of local information,

other sub-optimal reverse-prefixes are stored in the neighbourhood of the haplotypes that shares the longest reverse-prefix. The selection step is not performed in every marker, but only in a sparse subset of markers, since it is enough to capture long IBS states. Figure 3.1 illustrates the state selection on an example dataset.

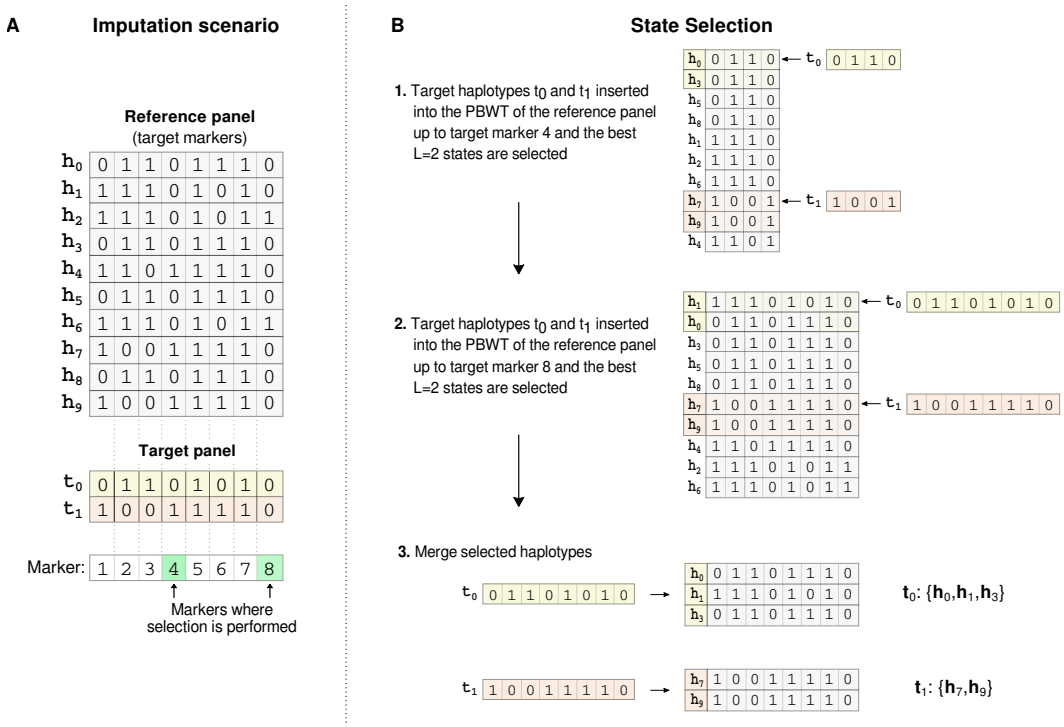

Figure 3.1: IMPUTE5 copying state selection (A) A reference panel of ten haplotypes $H = \{h_0, \ldots, h_9\}$ restricted to the set of target markers is shown together with a target panel of two haplotypes $T = \{t_0, t_1\}$. The copying state selection is only performed at a sparse subset of target markers (called selection markers). In this example, these are the 4th and 8th marker, and are shaded green. (B) The PBWT of the reference panel at target markers is built, one marker at the time, and each target haplotypes is located into the PBWT, using the rank operations. In (B1) target haplotypes $\{t_0, t_1\}$ are searched in the positional prefix array of the reference panel up to marker 4 and $L = 2$ reference haplotypes are selected for each target haplotype. In (B2) t_0 and t_1 are searched in the positional prefix array of the reference panel up to marker 8 and again L reference haplotypes are selected. The selected reference haplotypes are indicated by the shading colour (yellow for t_0 and red for t_1. (B3) The selected haplotypes are then merged to form the lists of copying states. The list may not necessarily be the same length. These states will be used in the HMM to perform imputation.

In order to locate the best reverse-prefix reference haplotype with each target haplotype, rank operations are used. For a target t at marker m, this search finds the location of the reference haplotype in the PBWT that shares the longest

reverse-prefix with t up to marker m.

The updated matched position p_{m+1} of the target t at marker m is given by rank operations in the right array (depending on the value of t_m) at previous value p_m:

$$p_{m+1} = \begin{cases} U_{p_m,m}, & \text{if } t_m = 0 \\ c_m + V_{p_m,m}, & \text{otherwise} \end{cases} \tag{3.2}$$

where c_m, U and V are respectively the number of 0s in each marker m, and the rank matrices for the reference panel at marker m. The list of locations is maintained for each target haplotype and updated at every marker.

The selection is performed every interval of length I (0.1 cM by default). These markers are called *selection markers*. The cost of the search is $O((N + K)M)$, where K is the number of target haplotypes, where $O(NM)$ is the cost to build the PBWT at target markers and $O(KM)$ is the cost of the selection algorithms.

Selection algorithms

IMPUTE5 implements two algorithms that select the "closest" L haplotypes from the position of the best reverse-prefix reference haplotype. We refer to them as *divergence selection* and *neighbour selection*. Both the algorithms have $O(LK)$ complexity, but the first requires to compute the divergence array from PBWT while the second does not.

The divergence selection algorithm has been proposed in the SHAPEIT4 method, and has the property of selecting the L haplotypes that share the longest reverse-prefix with the target haplotype. Since the PBWT encodes large amount of local information, the L longest reverse-prefixes are in the neighbourhood of the best matching haplotype found during the search. The divergence array contains all the information needed to select these states, exploiting Equation 2.5.

Recalling the notation introduced in the previous chapter, let us indicate as $A_{i,m}$ the positional prefix array at marker m at position i, meaning the i-th haplotype in the reverse prefix order up until marker m. Similarly, let us indicate with $D_{i,m}$ as the divergence array between the haplotype $A_{i,m}$ and $A_{i-1,m}$ up until marker m, indicating the biggest marker $m' < m$ such that haplotype $A_{i,m}$ and $A_{i-1,m}$ diverge.

Let's suppose we have the best match position p_m for haplotype t at marker m, obtained using equation 3.2. The algorithm only needs to check the values of the divergence array at position $i = p_m - 1$ and $j = p_m + 1$. If $D_{i,m} <= D_{j,m}$, then i is decreased and positional prefix $A_{i,m}$ is added to the list of selected states. Otherwise j is increased and $A_{j,m}$ is added to the list. The algorithm continues by comparing $D_{i,m} <= D_{j,m}$ until L states are selected. A pseudo algorithm of the copying state selection is shown in Algorithm 3.

The neighbour selection algorithm is an approximation of the divergence selection algorithm. Since the best L states are always in the neighbour of the best match position p_m, then, if we take a interval of size L centred at the best match position p ($L/2$ locations in both the directions), in the set of L states there are at least $L/2$ of the best matching reverse-prefixes. Taking this set of states requires less operations than the divergence selection, as it does not need to compute and interrogate the divergence array for that marker. In the case that the target haplotype is close to a border, less than L states are selected for that marker. A pseudo algorithm of the copying state selection is shown in Algorithm 4.

We tested another selection metric, based on the set maximal matchings (Durbin 2014). The set maximal matchings are the set of states that share the longest stretches with the target haplotype. We tested the use of the set maximal matches as a selection algorithm and we found that these contain a lot, but not all of the relevant information, and there is a small but evident loss in accuracy when we use only those matches. However, these states are promising and other algorithms, which incorporate set maximal matchings, can be proposed. Results of the comparison of different metrics are shown in chapter 4.

3.2.2 Input file formats

IMPUTE5 supports several input formats. It has support for indexed VCF or BCF file formats, which is particularly useful to quickly read reference haplotypes in particular region. The VCF/BCF file format is the standard format used to

Algorithm 3 Divergence selection algorithm.

Divergence selection at selection marker m starting from match position p

copy ← [ ]
copy.add($A_{p,m}$)
i ← p-1
j ← p+1
for *added* ← 1 to L **do**
    if $D_{i,m} <= D_{j,m}$ **then**
        copy.add($A_{i,m}$)
        $i \leftarrow i - 1$
    else
        copy.add($A_{j,m}$)
        $j \leftarrow j + 1$
    end if
end for

Algorithm 4 Neighbour selection algorithm.

Neighbour selection at selection marker m starting from position p

copy ← [ ]
copy.add($A_{p,m}$)
if $t_m == 0$ **then**
    $s \leftarrow max\{p - L/2, 0\}$
    $e \leftarrow max\{min\{p + L/2 - 1, N - 1\}, c_k - 1\}$
else
    $s \leftarrow min\{max\{p - L/2, 0\}, c_m\}$
    $e \leftarrow min\{p + L/2 - 1, N - 1\}$
end if
for $i \leftarrow s$ to e **do**
    copy.add($A_{i,m}$)
end for

represent a phased target panel of haplotypes and it is standard output of several phasing algorithms like SHAPEIT4.

The same file format can also be used to represent a reference panel of haplotypes, again taking advantage of the use of an index to read a specific chromosome region efficiently. However, because of the size of the reference panels and the typical decompression step needs to decompress gzipped input data, the reading time could dominate the whole running time of imputation algorithms, especially when imputing few individuals from a big reference panel. For this reason, we developed imp5 file format, a binary format designed to represent a set of haplotypes without

the need of an explicit decompression step performed by IMPUTE5. Details are provided in the following section.

imp5 File Format

We developed a new file format, called imp5 to read the reference panels quickly into memory. Imp5 uses a per-variant representation of the markers. This representation is dependent on the alternative allele frequency of the marker:

- if the alternative allele is rare (MAF $< 1/256$), the list of the indices of the haplotypes that carry the alternative allele are stored;
- if the alternative allele is non-rare (MAF $>= 1/256$), the sequence of alleles is stored using one bit per allele

Imp5 files are compact in memory and do not require other compression algorithms like gzip. This makes reading from a file an efficient operation, similar to BREF3 (B. L. Browning, Zhou et al. 2018).

This data structure is also used internally within IMPUTE5 to store the reference panel in memory. When imputing each target haplotype, at each target marker, the set of selected reference haplotypes that carry the alternate allele are needed. If the target site is stored as a bitset in the reference panel, then the lookup is straightforward. If the site is stored as a list of indices of alternate alleles, either the list of reference panel indices is searched for the selected state index, or vice versa, depending upon which search is likely to be quicker.

Another feature of the imp5 files is that they are indexed, so that regions can be extracted efficiently. The indexing was developed along the same lines as *bgenix* (Band and Marchini 2018), using sqlite3[18]. The indexing is an important feature for imputation, especially when imputing different windows on the same chromosome independently. Other file formats like BREF3 (B. L. Browning, Zhou et al. 2018) and m3vcf (Das et al. 2016) do not provide an index and therefore cannot directly interrogate arbitrary regions in constant time.

[18] `https://www.sqlite.org/index.html`

IMPUTE5 requires that the reference and the target panel files are indexed. In this way, several independent imputation jobs on different regions can be quickly read in memory and run at the same time, using a multi-process parallelization approach. A comparison of the memory requirements to store m3vcf, BREF3 and imp5 file formats is given in Table 3.1.

Reference panel	**vcf.gz**	**bcf**	**m3vcf.gz**	**bref v3**	**imp5**
Sim10K	0.06	0.04	0.02	0.02	0.04
Sim100K	0.83	0.56	0.15	0.11	0.42
Sim1M	18	10	1.90	0.75	4.06

Table 3.1: Memory (GB) required by reference file formats. Memory usage in Gigabytes required to store 10 Mb of reference sample data for 10 thousands (10K), 100 thousands (100K), and 1 million (1M) simulated UK European reference samples stored in vcf.gz, bcf, m3vcf.gz, bref3 and imp5 file formats. For the imp5 file format, the value reported is the sum of the memory required by the imp5 file, including the index file. Imp5 has been optimised to provide random access and fast reading time for a region of the chromosome, and not for data compression.

3.2.3 Output file formats and delayed imputation

The are several output file formats for imputation data that can be used by IMPUTE5. Other version of the IMPUTE software adopted the GEN file format, a textual representation of the genotype probabilities at each marker. Programs like QCTOOLS are designed to manage GEN files and perform standard filters of GWAS datasets. The GEN format is, however, not efficient to represent data for big cohorts of GWAS.

An alternative is to use VCF/BCF file formats as performed by other imputation methods. However, VCF/BCF file formats are designed to store genotype or haplotype data and not specifically for imputed data. To output genotype probabilities, the typical output of an imputation software, there is the need to add an additional field to store these probabilities. A more efficient way to represent imputed data is the new BGEN v2 file format (Band and Marchini 2018), a binary version of GEN files specifically designed for large imputed datasets, in which the probabilities are stored by using variable-precision packed bit representation and compression.

Previous imputation methods stored the genotype probability vectors for each target haplotype in memory, but with the advent of new generation reference panels, this data representation is no longer feasible, as reference panels contain hundreds of millions of variants.

IMPUTE5 addresses to this problem using *delayed imputation*, proposed by BEAGLE5 (B. L. Browning, Zhou et al. 2018). In the model, the classical forward-backwards calculations on the Li and Stephens model are performed using the reduced states space selected using the PBWT, resulting in very sparse probability vectors. IMPUTE5 stores state probabilities at consecutive markers for a reference haplotype if one of the state probabilities is greater than the inverse of the number of HMM states. These probabilities are defined only in the set of genotyped markers and not in the full set of markers of the reference panel. Since a small subset of the state probabilities at consecutive markers needs to be stored, imputation can be delayed during output, saving the memory required to store imputed probabilities at reference markers.

3.2.4 Parallelization

A typical IMPUTE5 job runs multiple regions of the same chromosome in parallel. Each region is completely independent of the others and can be run on different machines. The use of the indexing of the IMP5 files allows each process to read the reference panel efficiently.

Output is written in GEN, VCF, BCF or BGEN v2 file format (Band and Marchini 2018), explicitly designed to store imputed data. Merging GEN and BGEN files at the end of imputation is an efficient process and allows to impute each window independently (*cat* or *cat-bgen* commands are used to merge output files).

IMPUTE5 can also multi-thread each process. Multi-threading is developed using a shared memory approach. Each thread is responsible for a single target haplotype when running the HMM, or an imputation region between two target markers. The data sharing approach is crucial for reducing the memory required by each computational thread.

3.3 Real and simulated data experiments

We compared IMPUTE5 imputation accuracy, run-time and memory requirements against other methods using real and simulated reference panels. We used chromosome 10 and 20 from the 1000 Genomes Project (1000GP) (The 1000 Genomes Project Consortium 2015) and the Haplotype Reference Consortium (HRC) (McCarthy et al. 2016). We used a UK-European simulated data for 10K, 100K, and 1M samples generated using MSPRIME (Kelleher et al. 2016) as reference and target panel. Summary information of the reference panels used in this chapter are summarised in Table 3.2.

Reference Panel	**Mb**	**Number of Reference Samples**	**Number of Reference Markers**	**Target Marker Description**	**Number of Target Markers**
1000GP chr20	62.9	2,452	1,569,377	Omni2.5	53,183
1000GP chr20	62.9	2,452	1,569,377	OmniExpress-24	17,806
1000GP chr10	135.5	2,452	3,431,035	Omni2.5	111,570
1000GP chr10	135.5	2,452	3,431,035	OmniExpress-24	37,798
HRC chr20	62.9	31,470	884,983	Omni2.5	53,600
HRC chr20	62.9	31,470	884,983	OmniExpress-24	18,002
HRC chr10	135.5	31,470	1,927,503	Omni2.5	111,657
HRC chr10	135.5	31,470	1,927,503	OmniExpress-24	38,206
Panel A 10K	10.0	10,000	223,116	rand $>$ 5% MAF	3,333
Panel A 100K	10.0	100,000	747,162	rand $>$ 5% MAF	3,333
Panel A 1M	10.0	1,000,000	2,274,530	rand $>$ 5% MAF	3,333
Panel B 10K	10.0	10,000	223,116	rand $>$ 0.05% MAF*	33,333
Panel B 100K	10.0	100,000	223,116	rand $>$ 0.05% MAF*	33,333
Panel B 1M	10.0	1,000,000	223,116	rand $>$ 0.05% MAF*	33,333

Table 3.2: Summary of the real and simulated datasets used in comparing methods.
*in Panel B Sim1M

3.3.1 1000 Genomes Project

The 1000 Genomes Project was a landmark projects in human genetics, designed to provide a comprehensive description of genetic variation. The data have been used in a wide range of studies in human genetics. The 'phase 3' dataset contains phased sequenced data of 2,504 individuals sampled from 26 different populations

that can be included in five super-populations: African (AFR), American (AMR), European (EUR), East Asian (EAS), and South Asian (SAS). The dataset was generated using a combination of multiple sequencing approaches, including low coverage whole genome sequencing and dense SNP genotyping.

We used phased data available on the 1000 Genomes Project servers. We selected two random samples from each of the 26 populations for the study dataset and used the remaining data as a reference panel. For the reference panel, we removed monomorphic sites, which resulted in 3,431,035 markers on chromosome 10 and 1,569,377 markers on chromosome 20. For these results we used genomic build 37.

For the target panel, we masked data not present on the two considered SNP arrays. We used the very dense Illumina Omni2.5 array and the Infinium OmniExpress-24 v1.2. This resulted in 111,570 (Omni 2.5) and 37,798 (Infinium OmniExpress-24) target markers on chromosome 10, and 53,183 (Omni 2.5) and 17,806 (Infinium OmniExpress-24) target markers on chromosome 20.

3.3.2 The Haplotype Reference Consortium

The Haplotype Reference Consortium (HRC) (McCarthy et al. 2016) reference panel combines sequence data across 32,470 individuals from 20 sequencing studies with low-coverage WGS (from 4x to 8x coverage) of subjects with predominantly European ancestry.

We used data that was previously phased using SHAPEIT3. For our analysis, we randomly selected 1,000 target individuals from the HRC panel, and used the other 31,470 as a reference panel. We, again, restricted the data to markers having at least one copy of the minor allele and in the target samples and masked markers that were not on the Omni2.5 and Infinium OmniExpress-24 v1.2 arrays. This resulted in 111,657 (Omni2.5) and 38,206 (Infinium OmniExpress-24) target markers on chromosome 10 and 53,600 (Omni 2.5) and 18,002 (Infinium OmniExpress-24) target markers on chromosome 20.

3.3.3 Simulated reference panels

We simulated UK-European ancestry data using MSPRIME (Kelleher et al. 2016) in the same way as pereformed in the BEAGLE5 paper (B. L. Browning, Zhou et al. 2018). We simulated a 10 Mb region of sequence data for 11,000, 101,000 and 1,001,000 samples. We then extracted 1,000 samples from each of the three dataset, in order to have three reference panels of size 10K, 100K and 1M samples. We also split each of the 1,000 target samples into three different target panels of size 10, 100 and 1,000.

Since we cannot use real SNP array data, as there is no mapping of real genomic positions and simulated data, we masked all but 3,333 markers in the target panels, randomly selected between the markers having MAF $> 5\%$, to simulate chip sites. The reference panels have 223,116, 747,162 and 2,271,530 markers respectively. We refer to this setting (reference panel and target panels) as Panel A.

We created another setting, called Panel B, composed of 3 other simulated datasets having 1 million, 100,000 and 10,000 samples, but fixing the size to the same number of 223,116 markers. We, again, created three target panels of size 10, 100 and 1,000 samples at a subset of 33,333 markers by randomly selecting markers having MAF $> 0,05\%$ in the 1 million reference panel. Panel B is used to benchmark imputation on the same set of markers, varying the size of the reference panels.

3.3.4 Multi-chip HRC analysis

A previous analysis (Bycroft et al. 2018) compared the performance of imputation accuracy of several SNP arrays when imputing the same chromosome using the HRC reference panel. Specifically, the authors showed that the UK Biobank Axiom array gets similar imputation accuracy to the Illumina Omni2.5 array, which is a very dense chip and contains approximately three times the number of SNPs. This reinforced the validity of the UK Biobank Axiom array, specifically designed for imputation.

The study assessed imputation performance of several SNP arrays using data from 1000 Genomes Project by masking all the genotypes not present in the array. In particular, the authors used chromosome 20 data of 10 samples from CEU

population. All markers with a call rate below 90% were filtered out in order to only consider very reliable sites in the analysis. The study used data of 10 different genome-wide SNP arrays. The arrays used for the study are:

- Applied Biosystems UK Biobank Axiom
- Illumina 1M-Duo3_C
- Illumina HumanOmni5-4v1
- Illumina HumanCoreExome-12v1
- IlluminaGlobal Screening Array
- Illumina HumanHap300_v2
- Illumina HumanHap550_v3
- Illumina HumanOmni2.5-8v1
- Illumina Multi-Ethnic Global Array
- Affymetrix GenomeWideSNP_6

All the analysis were performed using IMPUTE4 and we replicated these results using IMPUTE5.

3.4 Results

3.4.1 Comparison of methods

We benchmarked IMPUTE5 against IMPUTE4, MINIMAC4 (v.1.0.0) (Das et al. 2016) and BEAGLE5 (v.16May19.351) (B. L. Browning, Zhou et al. 2018). A review of each method has been given in Chapter 1. We used default parameters for each program.

The IMPUTE methods require an explicit imputation and buffer region. We used an imputed region size of 5 Mb and 10 Mb for real and simulated datasets respectively with a buffer region of 500 kb.

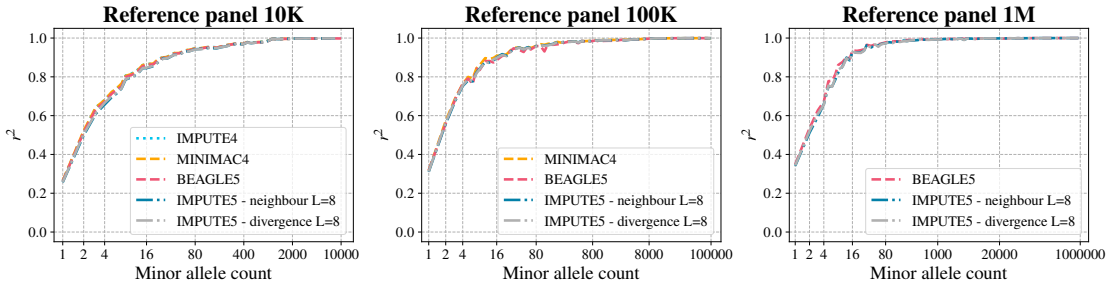

Figure 3.2: Imputation accuracy using Panel A dataset. Genotype imputation accuracy when imputing genotypes from a simulated reference panel of 10K, 100K and 1M UK-European reference samples (Panel A). Imputed alleles are binned according to their minor allele count in each reference panel. The squared correlation (r^2) between the true number of alleles on a haplotype (0 or 1) and the imputed posterior allele probability is reported for each minor allele count bin. The horizontal axis in each panel is on a log scale.

We used the HapMap2 (The International HapMap Consortium 2007) genetic map for BEAGLE5 and IMPUTE for real data imputation, and the true genetic map for analyses with simulated data. MINIMAC does not require a genetic map, as recombination parameters are estimated and stored when producing the m3vcf format input file for the reference data.

BEAGLE5, MINIMAC4 and IMPUTE5 use their specialised formats for reference panel data: BREF3 for BEAGLE5, m3vcf 4 for MINIMAC4 and imp5 for IMPUTE5. We tested IMPUTE5 selection algorithms using $L = 8$ and $L = 4$. IMPUTE4 was run with all reference panels except on the simulated reference panels, because IMPUTE4 is limited to 65,536 reference haplotypes and does not run on the two largest reference panels (100K, 1M).

We measured performance by comparing the imputed allele probabilities to the true alleles. Markers were binned using the minor allele count in the reference panel. We report the squared correlation (r^2) of the imputed and real imputed dosages.

All the computations were run on a 16-core computer with Intel Xeon CPU E5-2667 3.20GHz processors and 512 GB of memory.

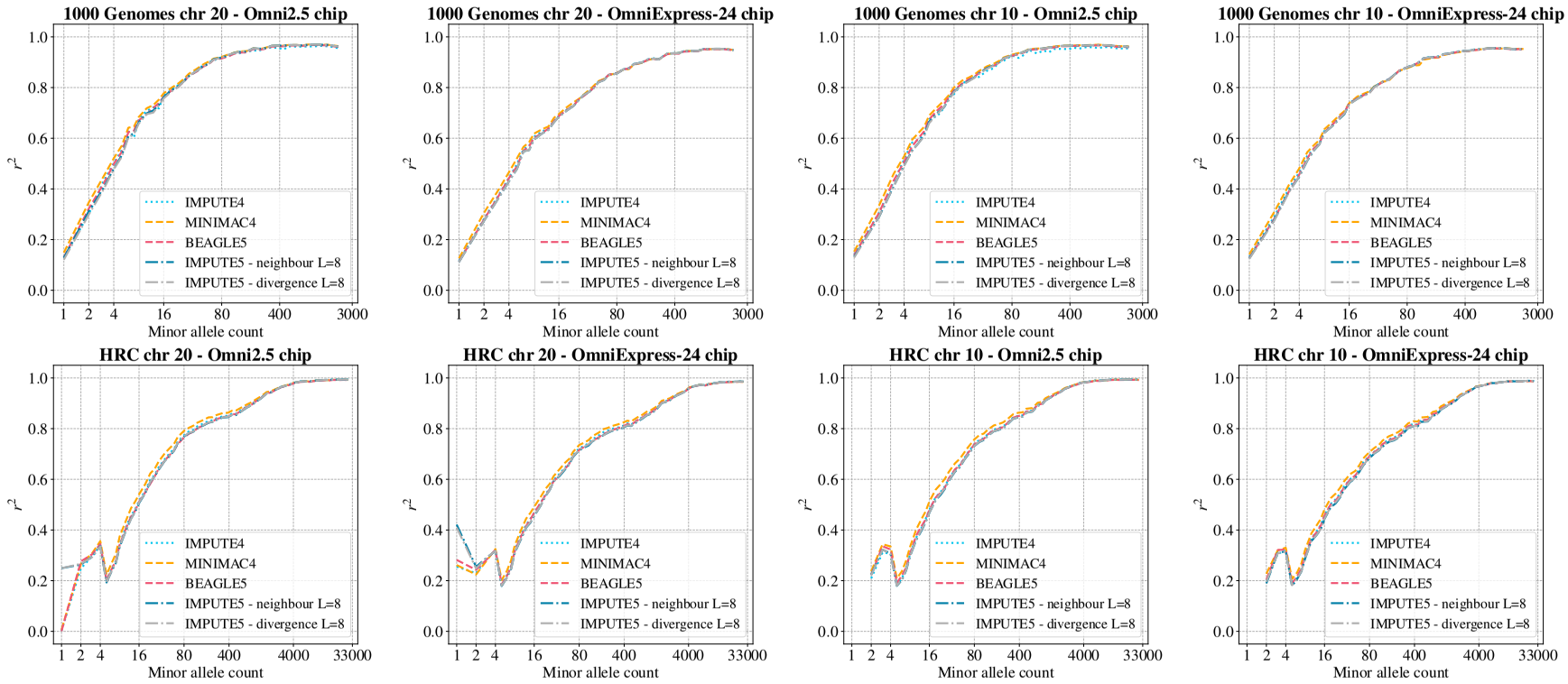

Figure 3.3: Imputation accuracy using 1000 Genomes Project and HRC dataset. Genotype imputation accuracy when imputing genotypes using the 1000 Genomes Project reference panel ($n = 2452$) and the Haplotype Reference Consortium reference panel ($n = 31470$). Imputed alleles are binned according to their minor allele count in each reference panel. The squared correlation (r^2) between the true number of alleles on a haplotype (0 or 1) and the imputed posterior allele probability is reported for each minor allele count bin. The horizontal axis in each panel is on a log scale.

3.4.2 Imputation accuracy

Imputation results for Panel A are shown in Figure 3.2. Panel A consists of three reference panels of size 10K, 100K and 1M samples. Results using IMPUTE5 $L = 4$ are shown in Figure A.3. There is a close agreement between the methods and this is explained by the fact that all the methods in this analysis use the same Li and Stephens probabilistic model (N. Li and Stephens 2003).

Imputation accuracy using different values of IMPUTE5 parameters are shown in Figure A.1. In particular, we tested the effect of the L parameter for a range of values ($L \in \{1, 2, 4, 8, 16\}$) and the two selection algorithms proposed (neighbour selection and divergence selection). Increasing the values of the L parameter has a positive effect in the imputation accuracy. Both the selection algorithms perform well for values of $L \geq 8$. However, neighbour selection algorithm seems to perform better for values of $L < 4$. This is explained by the fact that neighbour selection algorithm tends to select more states than divergence selection algorithm, making it more robust even with smaller values of L.

Imputation accuracy results for real datasets are shown in Figure 3.3. Results for IMPUTE5 $L = 4$ are shown in Figure A.4. There is a slight increase of the accuracy for the MINIMAC4 method that can be explained by the fact that MINIMAC4 uses the Baum-Welch algorithm to perform parameter estimation during the creation of the m3vcf files. This could also explain why there is no difference in accuracy for the simulated datasets, since the real recombination map in known and no genotyping errors are present in the reference and target panels.

3.4.3 Computational efficiency

Table 3.3 and Figure 3.4 show the per-sample computation times for IMPUTE5, BEAGLE5 and MINIMAC4 for 10 thousands (10K), 100 thousands (100K) and 1 million (1M) simulated reference panels when imputing a set of 1,000 target samples in a 10 Mb region on a single core. Results are plotted on log-log scale, which illustrates that both BEAGLE5 and IMPUTE5 exhibit sub-linear scaling as reference panel size increases. For Panel A results, moving from 10K to 1M

Dataset	Single core time (mm:ss)					
	Panel A			Panel B		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
10K	01:41	01:44	**00:48**	08:33	02:43	**02:41**
100K	09:20	03:50	**01:57**	65:15	04:10	**02:46**
1M	-	09:15	**06:22**	1690:50	18:08	**04:33**

Table 3.3: Single core time to impute Panel A and Panel B datasets. Total time to impute 1,000 target samples in a 10 Mb window using simulation data in Panel A and Panel B dataset. Time is shown using the format mm:ss. Bold font is used to indicate the method with the lowest time. MINIMAC4 was not able to run using the Panel A 1M reference panel due to time constraints in the construction of the m3vcf file.

reference samples increases the number of reference samples by a factor of 100 and the number of reference markers by a factor of 10, but IMPUTE5's imputation time increases by only a factor of 7.5. Overall, the results show that IMPUTE5 is consistently faster than all the alternative methods. Results using $L = 4$ are shown in Figure A.5 and Table A.2.

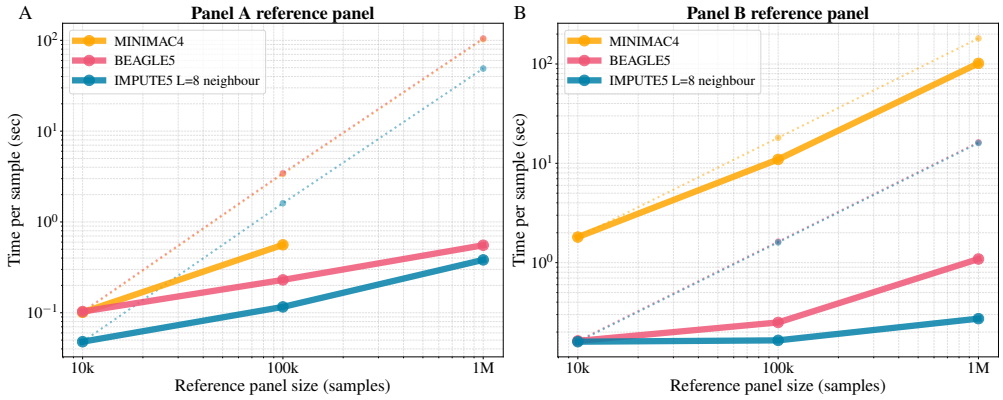

Figure 3.4: Per sample imputation time for Panel A and Panel B datasets. Per-sample CPU time when imputing a 10 Mb region from 10K, 100K and 1M simulated UK-European reference samples into 1,000 target samples using one computational thread. (A) Imputation time when using Panel A dataset (3,333 target markers). (B) Imputation time when using Panel B dataset (33,333 target markers). Axes are on log scale. Hypothetical linear scaling of MINIMAC4, BEAGLE5 and IMPUTE5 are shown as dotted lines, generated by projecting the time using the 10K reference panel. MINIMAC4 was not able to run using the Panel A 1M reference panel due to time constraints in the construction of the m3vcf file.

In Figure 3.4B we show computational time required to impute Panel B dataset. Panel B was created in such a way that all the reference panels (10K, 100K, 1M) have the same number of markers and the target panel has a very dense set

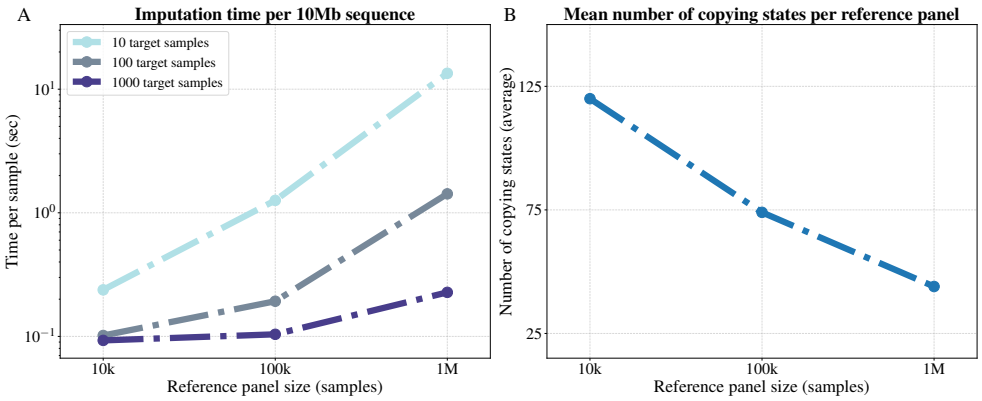

Figure 3.5: Sub-linear scaling using $L = 4$. (A) Time per sample spent to impute a marker in a 10 Mb region for reference panel size 10 thousands, 100 thousands and 1 million haplotypes, when imputing 10, 100 and 1000 target samples. (B) Mean number of copying states selected for the simulated reference panels. The number of selected states decreases by increasing the size of the reference panel, showing sub-linear scaling. Time and number of conditioning states are obtained with neighbours select and $L = 4$ using Panel B reference panel (33,333 target markers).

markers (33,333). We experimentally found that the time spent for the Li and Stephens calculations in the IMPUTE5 software actually decreases when increasing the number of reference haplotypes as a consequence of the reduced number of states retrieved. The reduced number of states retrieved is due to the fact that with more reference haplotypes, the copying state selection algorithm finds longer matching segments with the target, resulting in fewer states. The increase in time shown in Figure 3.4B from 10K reference panel to 1M reference panel is only due to increased time to read the input and run the selection algorithm, the only linear components of IMPUTE5.

Figure 3.5A shows that the imputation time per sample decreases when the number of target haplotypes increases This is mainly explained by the fact that typically, for a small number of target haplotypes, the PBWT construction is the main part of the selection algorithm and the copying states selection is a small fraction of the time.

Figure 3.5B shows the mean number of copying states selected across the 10Mb region for the imputation experiments using 10K, 100K and 1M simulated reference panels. It shows that, in IMPUTE5, the number of copying states decreases as the

number of reference haplotypes increases. This property is predicted from the Li and Stephens model Equation 1.12, that is itself an approximation to the coalescent model, whereby the probability of switching between copying states decreases as the number of reference haplotypes increases.

Dataset	**Single core memory usage (MB)**					
	Omni 2.5 chip			OmniExpress-24 chip		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
1000GP chr 20	1,792	7,243	**308**	1,696	6,698	**437**
1000GP chr 10	2,472	11,202	**367**	2,316	11,985	**522**
HRC chr 20	4,842	15,455	**2,368**	4,405	13,361	**936**
HRC chr 10	4,896	17,556	**2,428**	4,417	14,592	**1,001**

Dataset	**Single core time (mm:ss)**					
	Omni 2.5 chip			OmniExpress-24 chip		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
1000GP chr 20	02:33	01:24	**00:32**	01:37	01:11	**00:26**
1000GP chr 10	05:16	02:48	**01:04**	03:30	02:31	**00:52**
HRC chr 20	252:59	21:33	**09:13**	124:28	15:26	**05:52**
HRC chr 10	494:59	41:14	**18:26**	250:24	30:16	**12:01**

Dataset	**Parallel time (mm:ss)**					
	Omni 2.5 chip			OmniExpress-24 chip		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
1000GP chr 20	01:08	00:15	**00:03**	01:08	00:12	**00:03**
1000GP chr 10	02:25	00:28	**00:05**	01:56	00:24	**00:06**
HRC chr 20	27:13	02:12	**00:52**	15:07	01:20	**00:32**
HRC chr 10	56:08	06:07	**01:34**	30:54	02:41	**01:01**

Table 3.4: Memory usage and time to impute 1000 Genomes and HRC datasets. Memory usage and total time to impute a whole chromosome (chr 10 and chr 20) for 52 target samples when using the 1000 Genomes reference panel and 1000 target samples when using the HRC reference panel. Time is shown using the format mm:ss. Bold font is used to indicate the method with the lowest time.

Single core memory usage, time and parallel computation time of MINIMAC4, BEAGLE5 and IMPUTE5 ($L = 8$) are shown in Table 3.4. The table shows the

results of the three methods when imputing chromosome 20 and 10 for the real reference panels. The programs use different chunk sizes: IMPUTE5 run on chunks of 5 Mb, MINIMAC4 used chunks of 20 Mb and BEAGLE5 used chunks of 40 cM. For both the 1000 Genomes and HRC reference panel, IMPUTE5 is at least 2 times faster than BEAGLE5, and on the HRC reference panel it is over 20 times faster than MINIMAC4. Results using IMPUTE5 $L = 4$ are shown in Table A.1.

Multi-chip HRC analysis

We used the HRC reference panel to replicate a study performed in Bycroft et al. 2018, using IMPUTE5. The study consisted of the evaluation of different SNP arrays used data from 10 individuals in the 1000 Genome Project. For IMPUTE5, we used neighbour selection algorithm with $L = 8$. To account for the different numbers of variants of different SNP arrays, imputation performance is measured at the same set of variants when comparing chips.

Figure 3.6 shows the difference of imputation accuracy between the IMPUTE4 and IMPUTE5. The figure highlights a very similar imputation accuracy. This provides evidence that IMPUTE5 is a robust imputation method independent of the SNP array used. Imputation accuracy for different SNP arrays only using IMPUTE5 is shown in Figure A.6.

The run time of the the imputation for chromosome 20 was in the order of seconds for IMPUTE5 and 120 times faster than IMPUTE4. Since imputation was only performed on 10 target individuals, the two main reasons for this difference in the running time between the two methods are given by the big number of reference haplotypes present in the HRC reference panel and the fact that IMPUTE4 does not read an indexed and binary reference panel file formats and needs to scan the whole file to read the region of interest.

Effects of window size

IMPUTE5 does not have a default parameter for the window size. The method is robust to the choice of the window size, however increasing the window size would

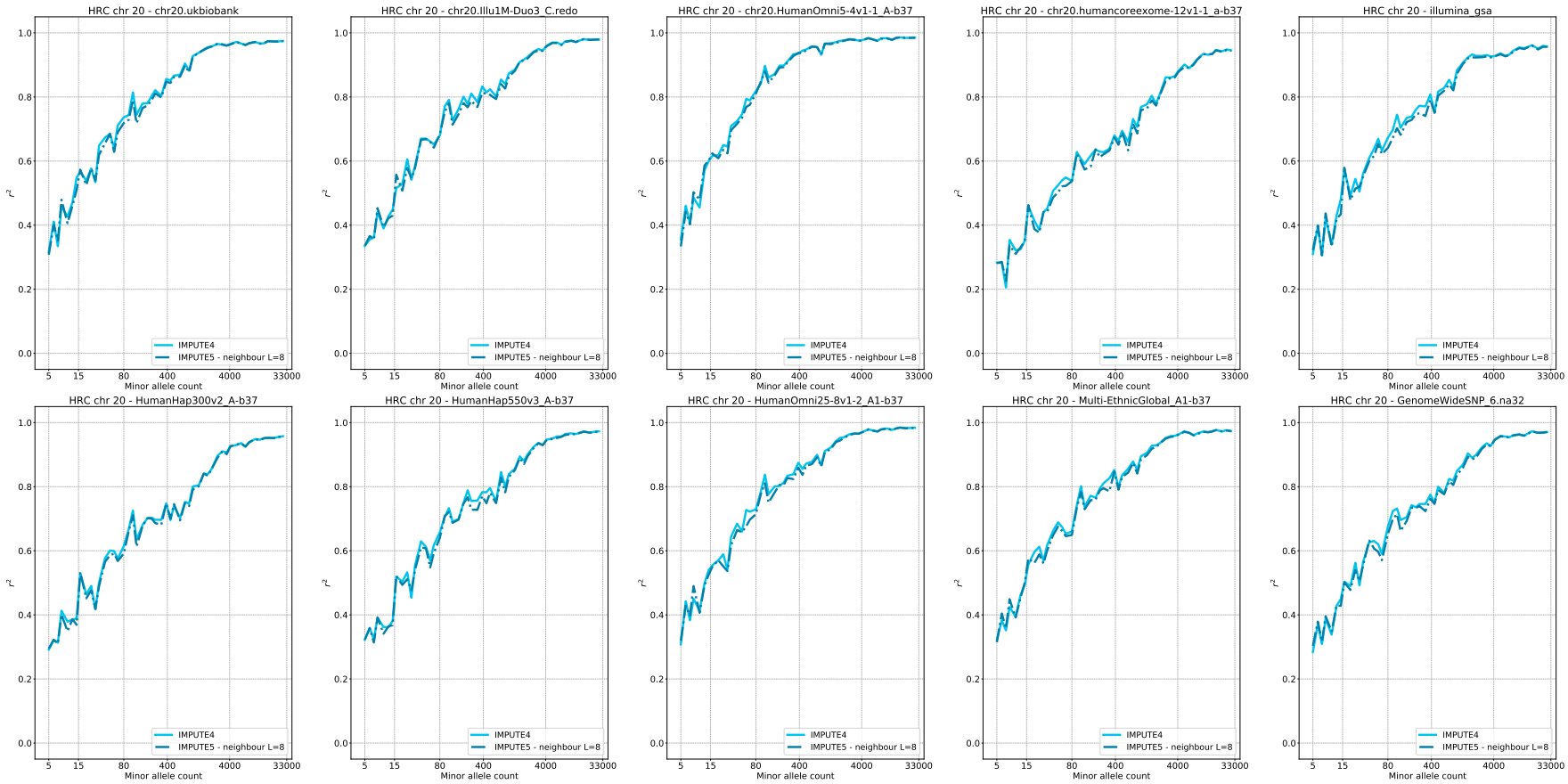

Figure 3.6: Imputation accuracy of different SNP arrays using IMPUTE4 and IMPUTE5 L=8 neighbour selection algorithm. Imputation accuracy for 10 different SNP arrays using IMPUTE4 and IMPUTE5. IMPUTE5 shows nearly identical imputation accuracy as IMPUTE4, showing that IMPUTE5 is a robust imputation method independently from the chip micro-array platform.

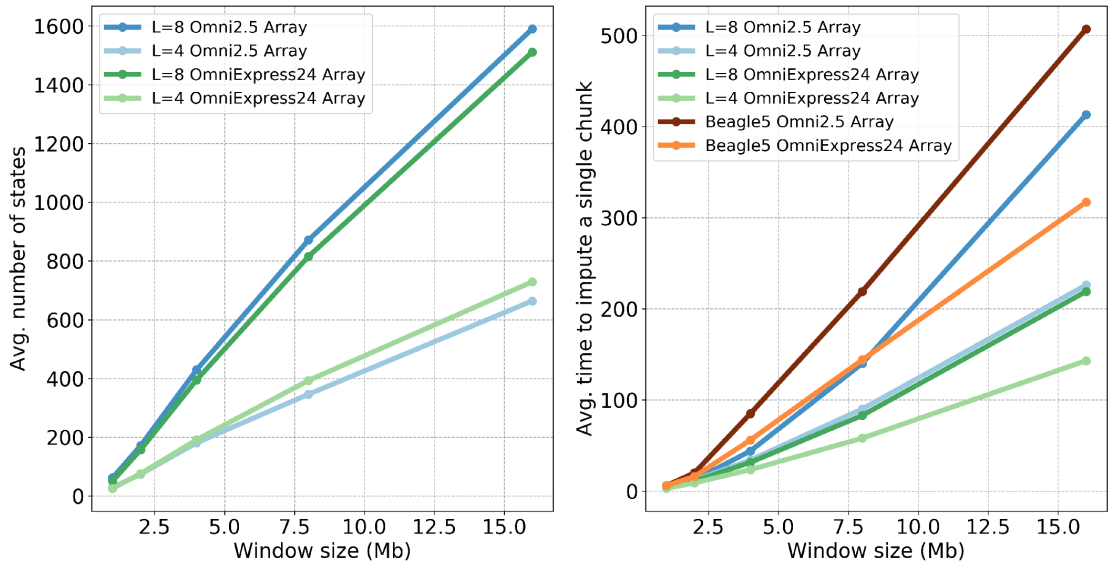

Figure 3.7: Average number of copying states and average running times varying the window size.

also increase the number of copying states retrieved for the selection algorithm, and therefore memory usage and computational time.

To verify the effect of the window size on the number of states retrieved and the running time of the programs, we run an experiment using the HRC reference panel. We imputed 1000 target stamples on chromosome 10 using data from the Illumina Omni2.5 and OmniExpress24 array, representing a high density and a medium density array, respectively. We used chunks having an average of 1,2,4,8 and 16 Mb imputation windows and we imputed using IMPUTE5 L=8, IMPUTE5 L=4 and BEAGLE5. We report the average number of states used by each of the IMPUTE5 configurations and time used by the three methods tested. Results are shown in Figure 3.7.

We note that the number of states retrieved depends heavily on the parameter L and only partially on the density of the chip (Figure 3.7, left). In terms of running time (Figure 3.7, right) IMPUTE5 is faster than BEAGLE5 even on big window sizes. Setting the parameter L=4 for big window size (e.g. 16 Mb) can be a good trade off in terms of accuracy and time, allowing fast imputation and leveraging long-range information.

Finally, as showed in Figure 3.5, by increasing the reference panel size, we would expect a decrease in the number of copying states, allowing to use bigger window size and bigger values of the parameter L, therefore increasing imputation accuracy.

3.5 Summary and discussion

In this chapter, we presented IMPUTE5: a method for genotype imputation designed to use big-scale reference panels. IMPUTE5 is the most recent of a series of the methods based on the IMPUTE model and provides new ideas and improvements.

We showed that the selection algorithm allows to retrieve an accurate subset of states that can be used during imputation. An important feature of the copy states selection algorithm is that the number of copying states decreases when the number of reference haplotypes increases. The haploid Li and Stephens HMM is then run on a sub-linear set of states. This allows to achieve a dramatic speed-up and impute using reference panel of million of haplotypes.

The development of a file format specifically designed for reference panel data (imp5) and the ability to read indexed input files allows quick imputation on a small region of the genome. Since reference panels are typically used for several imputation tasks, using a file format like imp5, increases the speed of all the subsequent imputation analysis.

Imputation will continue to have an important role in GWAS for at least the next few years due to the relatively cheaper cost of genotyping micro-array compared to WGS technologies. The advent of new sequencing projects like 100,000 Genomes Project or the UK Biobank will provide even more accurate and bigger reference panels for imputation.

The development of genotype imputation is still a fervent research area where method development is required in order to be able to handle the increased amount of data. The statistical models of genotype imputation has remained quite similar in the last decade, but the methods have become increasingly more efficient and are now able to use bigger and bigger reference panels, thus resulting in an increase in accuracy and number of variants of the imputed dataset.

4
State Selection Metrics

Contents

4.1 Introduction

The development of methods for haplotype phasing and genotype imputation relies on a similar statistic framework. With the increasing amount of available genetic information, in order to keep the computation requirements of these methods tractable with the currently available hardware, several approximations have been proposed during the years. In this chapter, we focus our attention on the state selection algorithms that have been developed for phasing and imputation, with the purpose of providing a framework to evaluate the accuracy compared to the full model.

The shared model used by imputation and haplotype phasing is the Li and Stephens HMM. By using the full Li and Stephens HMM as a gold standard, we derive the amount of probability explained by these approximations. We use the concept of Shannon entropy to explain how much uncertainty is present in the model, connecting high values in the information content to an increased amount of states needed to approximate the full model.

4.1.1 Basic definitions

We introduce some of the notation already used in the previous chapters. Let $H = \{h_1, h_2, \ldots, h_N\}$ be a set of N reference haplotypes, defined on M sites. Let $T = \{t_1, t_2, \ldots, t_K\}$ be a set of K study haplotypes defined at the same M sites.

We can run the Li and Stephens HMM on the haplotypes in T using H as a reference panel. For a target haplotype t_i, $(i = \{1, 2, \ldots, K\})$, we denote as $p_{nm} = Pr(Z_m^i = n|H, t_i)$ the probability that t_i copies the reference haplotype h_n, $(n = \{1, 2, \ldots, N\})$, at site m, $(m = \{1, 2, \ldots, M\})$, under the Li and Stephens model.

4.1.2 State selection metrics

An important contribution in the field of genotype imputation has been the IMPUTE2 model. The haploid version of IMPUTE2 is based on the haploid IMPUTE HMM. A key component of the model was to use only a highly informative subset of the haplotypes in the reference panel for imputation. IMPUTE2 requires a parameter k_{hap} indicating the number of haplotypes included in the copying set of the imputation region of interest. These k_{hap} haplotypes are defined to be those between the haplotypes in the reference panel that have the lowest Hamming distance with the target haplotype. The Hamming distance approximation works really well in practice, and this is the main reason that made IMPUTE2 one of the most successful imputation methods created so far. To this regard, the same idea of selecting k_{hap} haplotypes that minimise the Hamming distance has also been adopted by other successful methods for haplotype phasing, like SHAPEIT2.

The Hamming distance approximation comes with some limits. It has been shown that Hamming distance does not work well for rare variants and state selection algorithms which take local information into account can be more accurate (Huang et al. 2015).

Another selection metric that has been proposed is the set-maximal matching approach. A set-maximal matching between a target haplotype t_i and a reference panel of haplotypes H at marker m is defined to be the longest match that includes m between one or more haplotypes in H and t_i. It is possible to compute the list of all set-maximal matchings between t_i and H, for markers $\{1, 2, \ldots, M\}$, in linear time using the PBWT. The set of set-maximal matchings represent a compact and precise set of matches and can therefore be used for accurate phasing and imputation. However, this metric comes with a limit as well. It is true that the set of set-maximal matchings can represent a local optimal value, but all the other, informative and long matchings that are not set-optimal are discarded by the set-maximal matching set.

SHAPEIT4's divergence selection algorithm (introduced in chapter 3) has the advantage of taking into account both local information and maximal matchings. For this reason, it represents a compact and very accurate method for selecting copying states. The algorithm works by taking the L locally best haplotypes at each marker m. Its biggest limit is that, even if it is more flexible than the set-maximal matching approach, it could retrieve at the very best, only maximum L of the set-maximal matchings in a marker. For example, in a region with 12 identical set maximal matchings between t_i and H, this approximation is only able to retrieve L of them, where L is typically equal to 4 in the SHAPEIT4 software.

All the metrics have their own limits. An assessment of the Hamming distance, set-maximal and divergence selection algorithm is performed in the next section, using a dataset containing simulated data. The goal is to provide a deeper understanding of the current state selection metrics that has been an active research area for the last decade.

4.2 Methods

4.2.1 Shannon entropy

Recalling the notation introduced in section 4.1.1, we use p_{nm} to indicate the probability that t_i copies the reference haplotype h_n. At marker m, $p_{:m}$ represents a discrete probability distribution defined over the set of N reference haplotypes. Therefore, we can measure the uncertainty of probability at marker m using the Shannon entropy:

$$S_m = -\sum_{n=1}^{N} p_{nm} log(p_{nm}) \tag{4.1}$$

the entropy gives the average amount of information for an event in the probability distribution. Since the entropy is a measure of the information content, by taking its exponential, we obtain a quantity that can be interpreted as a measure of the minimum number of the states necessary to encode for the full probability distribution:

$$Q_m = e^{S_m} = e^{-\sum_{n-1}^{N} p_{nm} log(p_{nm})} \tag{4.2}$$

Most of the markers are well-explained using only few states. This means that the probability distribution at those markers has few peaks and is almost zero for all the others. Therefore there is less information content and thus obtains a small value for Q_m. However, if there are markers where there is more uncertainty, several states contribute to the probability distribution of the marker. In the extreme case, the distribution of the N states is uniform. In this case, there is the maximum information content and the value of Q_m is equal to N.[19]

Once the full Li and Stephens model has been computed, it is possible to estimate the information content from the distributions at each marker, which can be roughly interpreted as the minimum number of states necessary to represent the full probability distribution at the marker. When we are approximating the

[19] In the case of the probability is uniform Q_m can be written as $Q_m = e^{S_m} = e^{-\sum_{n-1}^{N} p_{nm} log(p_{nm})} = e^{-N \frac{1}{N} log(\frac{1}{N})} = e^{-log(N^{-1})} = N$

full Li and Stephens, markers with low information content are more easily well approximated than markers with more information content.

Regions with low information content have long stretches shared by only few haplotypes in the reference panel. Regions with higher information content are more complex and typically associated with higher recombination rates resulting in smaller shared stretches involving more haplotypes.

4.2.2 Test design

To evaluate the different state selection algorithms, we use the simulated dataset introduced in chapter 3, containing 10,000 reference samples and 100 target samples. Only 3,333 markers between those with MAF $> 5\%$ are considered. We run the full Li and Stephens HMM on each target haplotype, storing the full forward-backward probabilities. These probabilities represent the gold-standard of the approximations considered. One of the values we report is the amount of probability explained by the approximations in a marker, which is obtained by the sum of the probabilities in the full model restricted to the states present in the approximated subset of states.

By increasing the parameter L in the divergence algorithm, an higher the number of states is retrieved, resulting in an increase in the amount of probability explained by the approximation compared to the full Li and Stephens increases. For this reason, we tested different values of the L parameter for the divergence and the neighbour algorithm. The values used are $L = 1, 2, 4, 8, 16, 25, 100$.

The set-maximal set selection does not require parameters and the number of copying states is only dependent by the data. The Hamming distance metric requires the parameter k_{hap}, and we set this value to the number of sates retrieved using the divergence select algorithm for the different values of L.

In order to compare the accuracy of the real state selection metrics with two extreme cases, we also test two other metrics. The random states metric selects k_{hap} haplotypes uniformly randomly chosen from the set of states and should represent a lower bound of all the selection algorithms. We also include the best states metric. This is defined as the k_{hap} haplotypes in the reference panel that maximise the

amount of probability explained at the marker. Conversely, this represents the upper bound of all the selection algorithms having k_{hap} markers. In our tests, we set k_{hap} to be the number of states retrieved using the divergence select algorithm for both of these metrics, for different values of the parameter L.

We evaluated the different selection metrics at each of the 200 haplotypes in the study panel. We then aggregated the data of different haplotypes and different markers obtaining the average amount of probability explained by each of the selection algorithm using different values of L (and then k_{hap}).

4.3 Results

4.3.1 Comparison of selection algorithms

We firstly examined data at the haplotype level. Figure 4.1 shows data of one of the 100 haplotypes in the study. It shows the amount of probability explained at each marker using the three selection algorithms. For the divergence and neighbour selection algorithms different values of the parameter L are used, $L = 1, 2, 4, 8, 16, 25, 100$, where the number of states used is ranging between 52 ($L = 1$) and 6,324 ($L = 100$) for the divergence selection and 65 ($L = 1$) and 6,892 ($L = 100$) for the neighbour selection algorithm. The results show that, for small values of L, markers with small information content are typically well approximated, with a high amount of probability explained even for $L = 2$ or $L = 4$. It is also evident that neighbour selection algorithm needs more states (increased value of L) to reach the same amount of probability explained.

The set maximal matching selection retrieves 61 states and gets high values of probability explained in certain regions. For example the first peak at the left of the plot is retrieved by the divergence algorithm only when using $L = 8$. This suggests that a combination of the divergence selection and the set-maximal matching metric could increase the amount of explained probability for low values of L.

We also aggregated data of 100 target samples obtaining the average amount of probability explained for each metric. In this context we also included the hamming distance selection, the best states and random states metrics. These three metrics

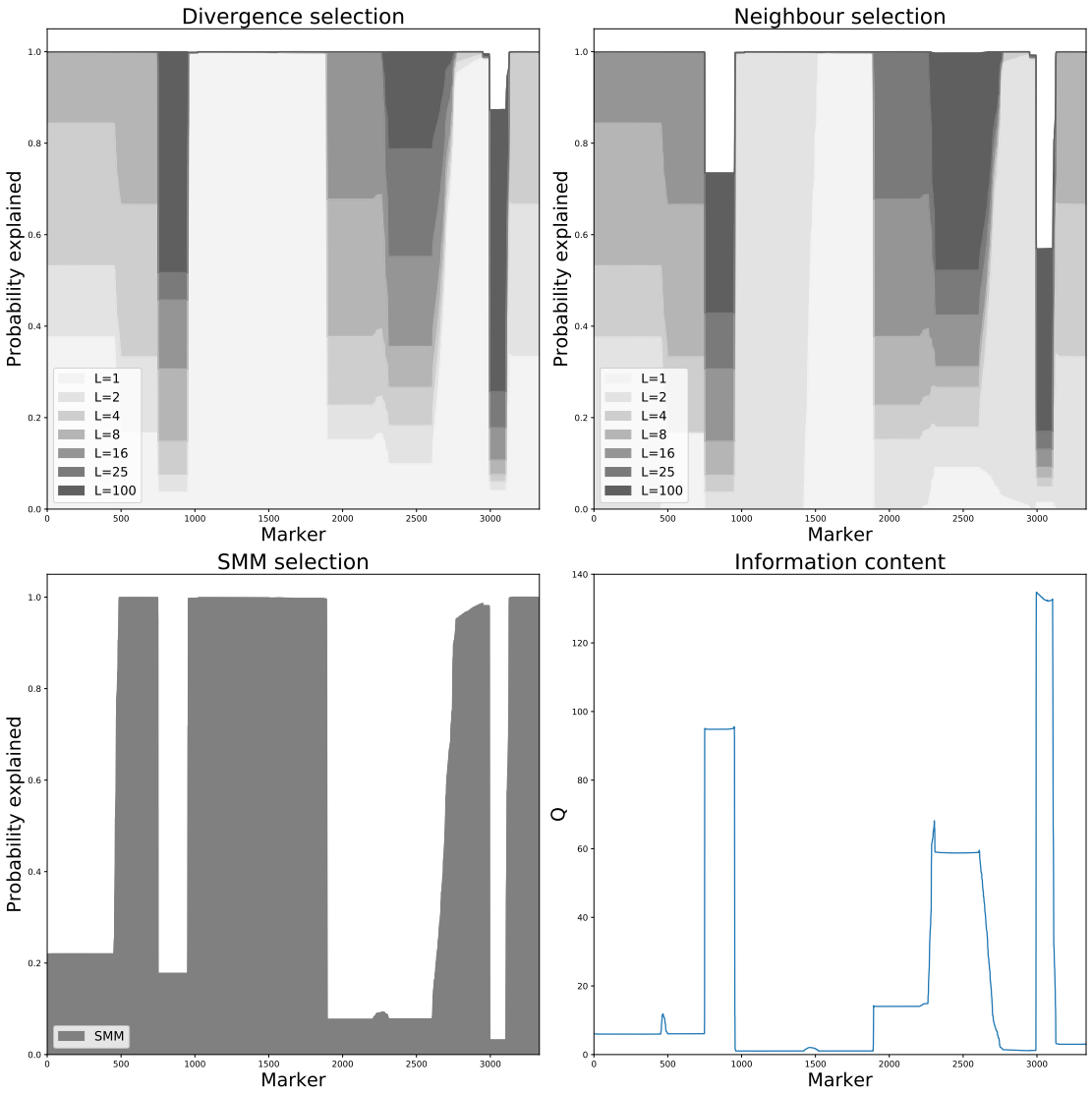

Figure 4.1: Probability explained by different state selection algorithms for a target haplotype. Amount of probability explained at each marker of a target haplotype using different selection algorithms and relationship with the information content Q_m. Divergence selection algorithm (top-left) is more accurate than all the other methods. Larger values of the parameter L retrieve more states in the set, increasing the overall probability explained. The neighbour selection (top-right) is an approximation of divergence selection and it shows similar but reduced accuracy. The set-maximal matchings selection (bottom-left) retrieves a number of states similar to the previous two selection algorithm when using $L = 1$ and it is more accurate. For example, the first peak on the left of the SMM is taken in the divergence selection only when $L = 8$. It does, however, leave out part of the information, especially for complex regions having high information content (bottom-right).

need a parameter k_{hap} indicating the number of states in the set, and we set this value to the number of states in the divergence array selection for different values of L.

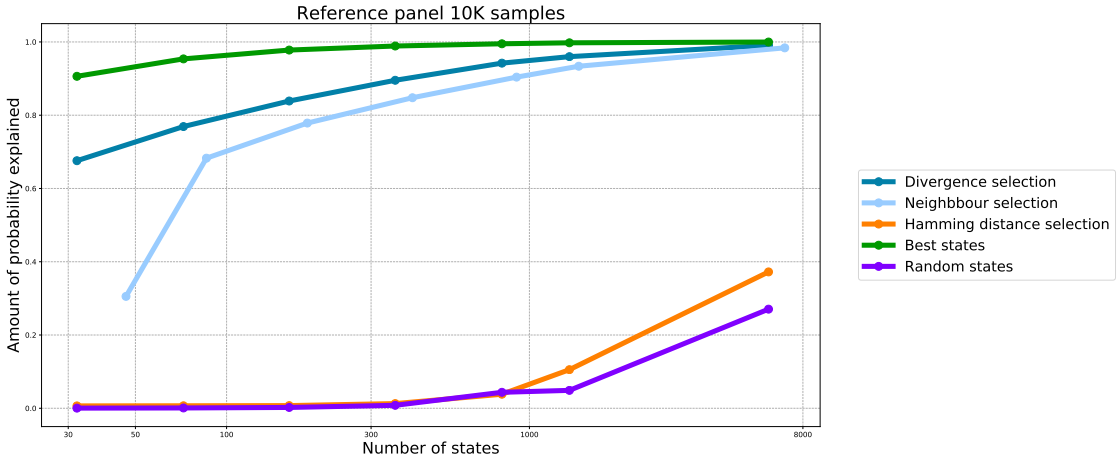

Figure 4.2: Aggregate probability explained by different state selection metrics for 100 target samples. Aggregate amount of probability explained by different state selection metrics for 100 target samples using a reference panel of 10,000 samples. Different points represent values of $L = 1, 2, 4, 8, 16, 25, 100$ or k_{hap} (defined as the number of states of divergence selection varying L). The divergence selection algorithm and the neighbour selection approximate well the full Li and Stephens model. We found that the Hamming distance metric is only slightly better than the random states selection.

The divergence selection algorithm seems to approximate well the best states even for small values of L. Similarly the neighbour selection algorithm is a good approximation of the divergence selection and this motivates its use for imputation. It typically requires an increased amount of states compared to the divergence selection, but the amount of probability explained is very similar, especially for values of $L > 2$.

It is surprising to notice how the Hamming distance selection has similar performance to the random selection algorithm. This might be due to the fact that the region considered might be too big for Hamming distance methods.

4.4 Summary and discussion

In this chapter we evaluated different selection metrics using the full Li and Stephens model as a gold-standard. We used simulated and real datasets to compare the amount of information loss when using a state selection algorithm.

Results reinforced the idea that the divergence selection algorithm represents a good approximation of the full Li and Stephens model. We also noticed that it

would be possible to include information from the set-maximal matching metric in order to increase the accuracy for small values of L.

The divergence selection algorithm is run in the forward direction. This produces a set of similar, yet different, set of states than the same algorithm run in the the opposite direction. This is because the divergence selection is completely dependent on the reverse-prefixes. A suggestion could be to integrate SMM and divergence selection from both the directions.

A problem of the divergence metric is that it is dependent on the value L fixed for all the markers. It can be possible to get an approximation of the information content of the marker by simply observing the values of the divergence array for both the forward and backward direction. In this way, it would be possible to adapt L to the information content of the marker.

The state selection algorithms based on the PBWT, presented in this chapter, are linear in the number of haplotypes and markers. It can potentially be possible to reduce this cost, getting linearity only in the number of markers. This represents an ideal scenario, as it requires the PBWT and the FM-index to be already computed for the set of markers of interest (e.g. markers present of a SNP array). However, since the commercial SNP array markers are known, it may be reasonable to represent common variants in the form of a PBWT in order to allow fast query and state selection.

5

Chromosome painting using the IMPUTE5 model

Contents

5.1 Introduction

Methods for tracking ancestry between individuals are based on the concept that long shared stretches denote recent common ancestry. Several methods in population genetics exploit this concept such as haplotype phasing, genotype imputation, identical by descent (IBD) detection and chromosome-painting methods.

The aim of chromosome painting is to infer the ancestry along chromosome segments. It usually involves the modelling of a sample haplotype as a mosaic

composed of chromosomal segments, chosen from a set of ancestral haplotypes. A natural choice of chromosome-painting algorithms is the Li and Stephens model that explicitly takes into account mutation events and recombination hotspots. From the Li and Stephens model it is possible to obtain the probability for a sample haplotype copying from each of the reference panel haplotypes at every considered marker. Several chromosome-painting methods have been developed in the last decade (Price et al. 2009; Lawson et al. 2012; Hellenthal et al. 2014).

In this chapter we propose a method to cluster haplotypes using aggregate information of haplotype copy probabilities, obtained within the IMPUTE5 environment, to evaluate the amount of shared ancestry between different populations. We propose to use the same dataset as a target and a reference panel of haplotypes and running the Li and Stephens model on each target haplotype on the full set of reference haplotypes, but banning the haplotypes in the reference that correspond to the target.

We test the accuracy of the IMPUTE5 state selection metric by measuring the expected sequence sharing between populations in the 1000 Genomes project (The 1000 Genomes Project Consortium 2015). The 1000GP contains 2,504 WGS sequences from 26 populations that can be grouped in five super-populations: Africans (AFR), Americans (AMR), Europeans (EUR), East Asians (EAS) and South Asians (SAS).

We find evidence of the population structure for common variants using both the full and the approximated Li and Stephens model. Clusters of populations are linked to the real ancestry of the groups. We also verify that the amount of information loss by using the copying state selection algorithm is minimum compared to the full model.

Finally, we measure the imputation accuracy of both the models. We show that IMPUTE5 neighbour selection algorithm has the same performance as the full Li and Stephens model, but several times faster even for a reference panel of modest size as the 1000 Genomes Project.

5.2 Methods

5.2.1 Expected sequence sharing

Recalling the definitions of section 4.1.1, we split the sequence t_i into chunks centred on each of the M loci, having boundaries equidistant from each of two consecutive markers . The expected amount of sequence (in cM) copied by t_i from h_n along all the sites is:

$$d_n = \sum_{m=1}^{M} q_{nm} \tag{5.1}$$

where:

$$\begin{aligned} q_{nm} &= \frac{p_{nm}(g_{m-1} + g_m)}{2}, \text{ for } m \in \{2, \ldots, M-1\} \\ q_{n1} &= \frac{p_{n1} g_1}{2} \\ q_{nM} &= \frac{p_{nM} g_{M-1}}{2} \end{aligned}$$

and g_m is the genetic distance in cM between the mth and $(m+1)$th sites. For each haplotype t_i, there is then a vector $D_i = \{d_1, \ldots, d_N\}$ representing the expected amount of copied sequence from each haplotype in the reference panel.

5.2.2 Clustering

We propose to use a reference-target framework to cluster haplotypes. The idea is to use the same set of haplotypes as a target and reference dataset for the copying state selection and the Li and Stephens HMM. For each target haplotype, we run the selection algorithm (e.g. neighbour selection) to select states from the reference, and then run the Li and Stephens HMM.

Since the reference and target datasets are based on the same samples, we ban from each target haplotype's copy set the two haplotypes of the reference panel that correspond to the target sample. In this way, the reference panel has size $N-1$ samples, dynamically computed from the dataset of size N.

The full clustering algorithm for a set of N samples, without state selection, runs the Li and Stephens model on the full set of haplotypes, except two banned

haplotypes, and has a computational cost proportional to the number of samples in the dataset (N) and the number of genotyped markers in the reference panel (M), in total $O(N^2 \times M)$.

By using a state selection algorithm, we reduce the complexity of the Li and Stephens model by introducing a cost for the state selection. If M_u denotes the number of sparse markers where the selection is performed and S_t denotes the number of copying states for target sample t then complexity is:

$$O\left(NM \times (NM_u)\right) + O\left(N \times \sum_{t=1}^{N} S_t M\right) \tag{5.2}$$

In practice, the time spent for the selection algorithm is a small fraction of the time used for the Li and Stephens model.

Computing the copying states from the full reference panel of size N samples and then ban the target sample from the set is not the same as selecting the copying states from the reference panel without the target sample (having size $N - 1$). However, we note that the set of states retrieved using the divergence state selection algorithm, on the banned dataset, is an equivalent set (in terms of length of the reverse prefixes retrieved) as the states obtained if the actual reference panel of size $N - 1$ is constructed. This is not true for the neighbour state selection algorithm, but it is reasonable to think to this difference as minimum.

5.2.3 Imputation testing framework

One feature of using the same dataset as target and reference panel is that the dataset can potentially have a different number of markers, for example, the reference panel contains WGS data and the target is the same dataset, but with non-chip markers masked. In this way, it is possible to perform imputation and evaluate imputation accuracy only by using a single dataset.

5.2.4 Test design

We used the publicly available 1000 Genomes Project (1000GP) to verify the quality of the copying states algorithm of the IMPUTE model. We used data of

chromosomes 1 to 22 and masked data in 1000GP not present in the Illumina HumanOmni2.5-8v1 array. The 1000GP has been treated as both the reference and a target panel. Since the target and the reference panel are based on the set of individuals, we needed to avoid copying from the same sample and running the HMM on all the others. We banned the reference haplotypes corresponding to the target samples using the IMPUTE5's *ban-hapid* flag. This resulted in a reference panel composed of 2,503 samples (5,006 haplotypes).

We developed an IMPUTE5 option specifically to compute and output the list of vectors D_i for all i in T. The program was run with two different settings, using (i) all the reference panel states and (ii) the neighbour selection algorithm with $L = 8$. The output of (ii) assumes that all the states not selected from the IMPUTE5 neighbour selection algorithm have $D_n = 0$. We compared the concordance of the two IMPUTE5 modes, and verified that the copy states probabilities follow the population structure.

We also run imputation for all the 2,504 samples using the same masked dataset as a target panel and the full dataset as a reference panel, again banning reference haplotypes in the copy state selection that belong to the target haplotype. We report the running time of the two settings (i) and (ii) and imputation accuracy in terms of the r^2 correlation of true and imputed dosages across all the 22 chromosomes.

5.3 Results

We run the IMPUTE5 model on the whole set of 2,504 samples belonging to 1000 Genomes Project using the same dataset as a reference panel. The output was the vector D_i for all the haplotypes in 1000GP describing the expected amount of copied sequence from all the other sequences in the panel. We fixed the expected amount of sequence copied for sample i to sample i to zero by default. We excluded the possibility of sample i to copy from itself during the HMM. By aggregating data across all the chromosomes in the study, we have a global D_i for each target sample i.

Figure 5.1 shows the distribution of the vector D_i for each target sample by indicating by a dark blue dot a high value in D_i. Haplotypes of the 1000 Genomes

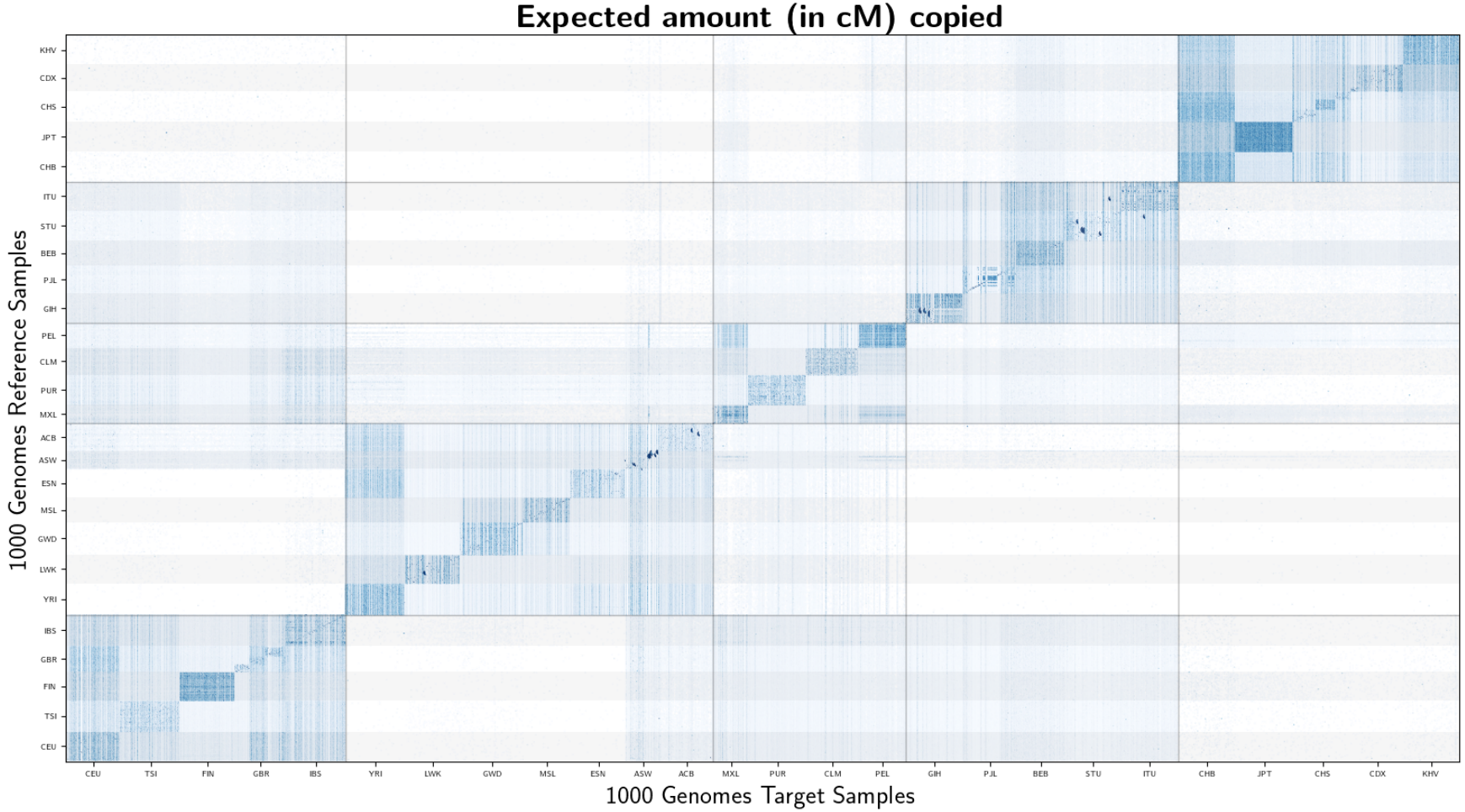

Figure 5.1: Graphical visualisation of the expected amount of sequence copied from each population for all the samples in 1000 Genomes Project. Results are obtained using IMPUTE5 neighbour selection $L = 8$. Samples are grouped by populations and super-population groups. From left to right EUR, AFR, AMR, EAS and SAS are shown. Darker blue dots indicate a bigger amount of sequence shared between two individuals. Clusters of all the populations (small blue square patterns) and the super-populations (blue square patterns) appear on the diagonal line. This indicates that most of the expected copied sequences derive from samples of the same population, due to more recent common ancestry.

project represent the x an y axis of the plot, indicating the target and the reference haplotypes, respectively. As expected, we can identify patterns of the populations and super-populations, due to a recent common ancestry and longer shared sub-sequences between samples in the same population (and super-populations). Two extreme cases are Finnish (FIN) and Japanese (JPN) populations, where almost all the expected sequence sharing comes from members of the same population, indicating strong genetic structure.

5.3.1 Effects of the selection algorithm

In addition to the aggregation of information across all the chromosomes, we have also aggregated data by populations in order to evaluate the amount, on average, that samples in a population copy sequences from other populations. Intuitively, we expect that members of the same population tend to have longer shared stretches, due to a more recent common ancestor, and, therefore, bigger values of the expected sequence sharing.

We performed this analysis using IMPUTE5 with the neighbour selection algorithm ($L = 8$) and IMPUTE5 with all the states, in which no selection is performed. IMPUTE5 with all the states represents the full Li and Stephens model, used by chromosome painting algorithms.

Figure 5.2 shows the aggregated expected sequence sharing of different EUR populations using the two methods. There is good concordance between the methods, suggesting that the selection algorithm has a similar power as the full model. Each population has the longest shared sequence with its own population. This is particularly evident for Finnish population, having strong genetic structure. Results regarding African, American, South Asian and East-Asian super-populations are shown in Figures B.5, B.6, B.7 and B.8.

Values of the total shared sequence that individuals share to each population is affected by the small sample size of individuals belonging to each population in the experiment. On short segments we can note a non negligeble amount of sequence sharing between individuals belonging to unrelated populations. This

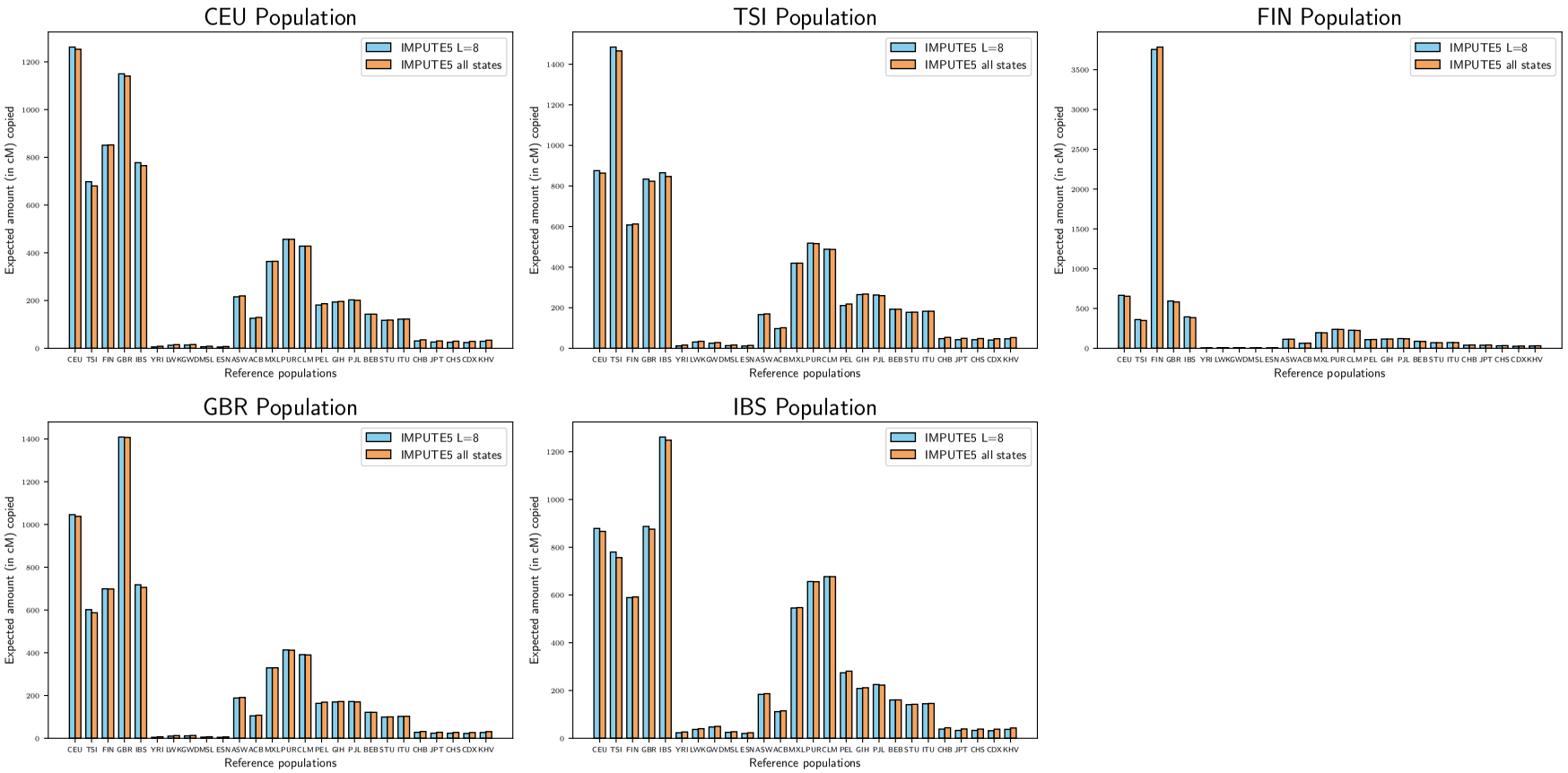

Figure 5.2: Average expected amount of sequence copied from each of the 1000 Genomes populations for the five EUR populations. In a population (e.g. CEU) all samples are grouped together and the average of the distance copied from all the populations is computed. Here we show expected amount of distance copied using IMPUTE5 neighbour select algorithm and IMPUTE5 all states. As we can see there is very good concordance between the two populations. Each population tends to copy more from members of their own population and in general from members of the super-group they belong to (EUR), due to a more recent common ancestor. In the case of the FIN population, the peak indicates a very segregated population, with less shared ancestry with others.

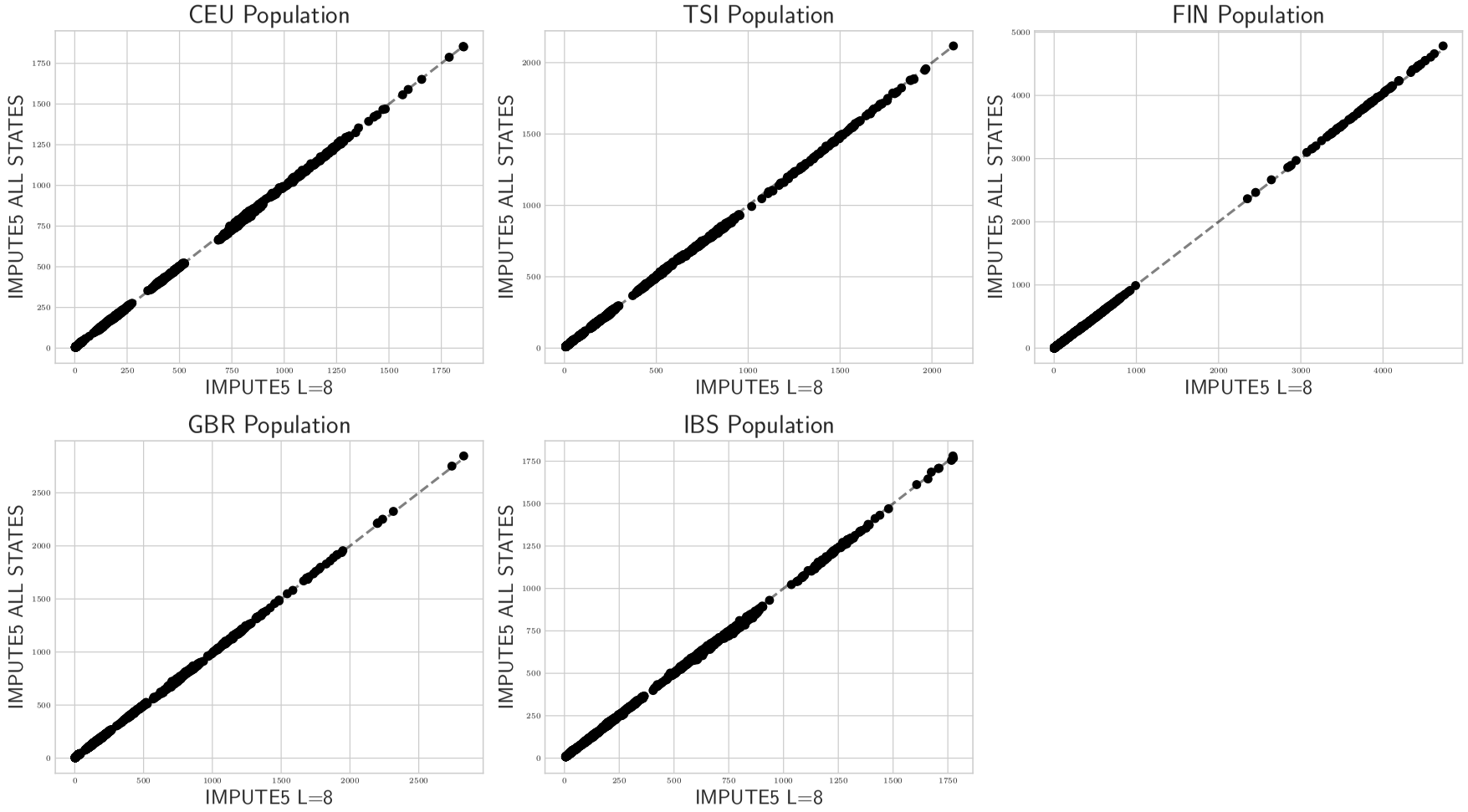

Figure 5.3: Scatter plot of the concordance between values of expected amount of sequence copied from the reference panel (D_i) of IMPUTE5 neighbour select and IMPUTE5 all states for EUR populations. All the samples belonging to a population are grouped, and the expected values for the all the EUR population is shown. There is visible concordance between the two IMPUTE5 settings.

can be seen for each population in Figure 5.2, for example from TSI population to JPT population. Increasing the sample size of our experiment we would be able to find longer matches from the same population, reducing the amount of sharing between unrelated populations.

We also note that IMPUTE5 with the neighbour selection algorithm ($L = 8$) tend to be more confident of copying within the same population (and even super-population) then IMPUTE5 with all the states. This is due to the fact that IMPUTE5 with the neighbour selection algorithm gives a value of zero shared distance for all the states non in the copying states and thus slighly increasing the amount of shared distance of the individuals within the copying states.

Figure 5.3 shows the concordance between the two methods by plotting the expected sequence sharing between between each individual in a population and all the populations in the study obtained using both the methods as x and y axis. It is evident that there is almost perfect concordance between the methods even at the sample level. This means that IMPUTE5 selection algorithm does not change consistently the accuracy, compared to the full Li and Stephens HMM and can be used instead of the full model. Results regarding other populations are shown in Figures B.1, B.2, B.3 and B.4.

5.3.2 Imputation

We imputed the 2,504 samples using IMPUTE5 neighbour select algorithm $L = 8$ and IMPUTE5 using all the states. In the second setting, no time is spent on the selection algorithm, but the HMM is run on a much bigger set of states. We run one method at the time using a maximum of 16 parallel processes per chromosome.

Figure 5.4 shows the imputation accuracy of both the IMPUTE settings used for chromosome 20 and chromosome 10. As expected, the imputation accuracy is the same for both the methods, providing another validation of the IMPUTE5 state selection algorithm.

The total running time of IMPUTE5 using neighbour selection algorithm ranges from 3.5 to 4 times faster than IMPUTE5 using all the reference panel states.

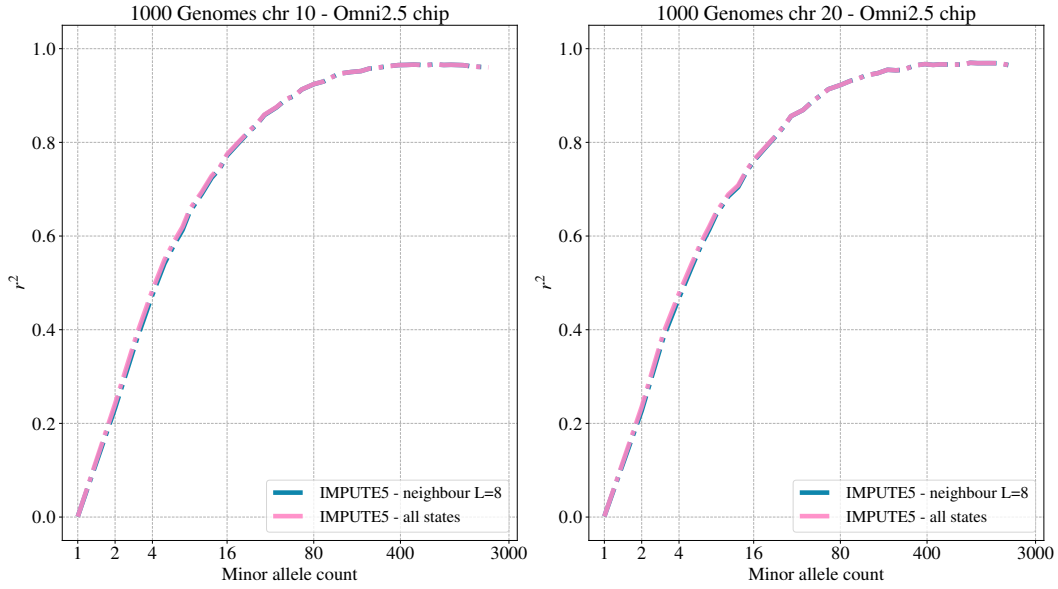

Figure 5.4: Imputation accuracy (r^2) for chromosome 10 and 20 using IMPUTE5 neighbour selection algorithm and IMPUTE5 using all the states. Markers are binned using the MAC in the reference panel.

In addition, using the state selection algorithm results in using approximately 2.5 less memory. Results for all the chromosomes are shown in Table 5.1. The running time of the selection algorithm is approximately 0.2% of total running time for a reference panel of this size.

These results suggest that even for relatively small reference panels like 1000 Genomes Project, the use of IMPUTE5 with neighbour selection algorithm reaches the same accuracy as using all the states but using much less computation time and memory. For reference panels of hundreds of thousands of haplotypes, using all the reference panel states would not be feasible and the use of the state selection algorithm is necessary to run imputation on currently available hardware.

5.4 Summary and discussion

In this chapter we evaluated the painting information of the IMPUTE5 model when using the all the copying states of the reference panel and when the neighbour selection algorithm is used. To do this we defined a simple metric, the expected sequence sharing, to quantify the amount of similarity between two haplotypes.

Chr	Chunks	Method			
		IMPUTE5 NS L=8		IMPUTE5 ALL STATES	
		Time (hh:mm:ss)	Memory	Time (hh:mm:ss)	Memory
1	16	00:18:16	17.6	01:24:34	40.3
2	16	00:19:31	18.2	01:30:40	42.1
3	16	00:15:16	15.1	01:17:18	35.2
4	16	00:14:59	14.1	01:13:46	34.6
5	16	00:13:14	13.2	01:08:33	32.1
6	16	00:13:58	14.1	01:17:40	33.3
7	8	00:15:22	20.4	00:59:42	45.8
8	8	00:13:28	19.6	00:58:02	43.4
9	8	00:11:28	18.4	00:48:46	40.3
10	8	00:13:12	20.2	00:53:27	41.3
11	8	00:12:13	20.0	00:54:26	41.1
12	8	00:12:59	20.5	00:51:48	42.8
13	8	00:08:42	15.5	00:39:34	35.8
14	8	00:07:47	14.7	00:37:36	34.3
15	8	00:08:03	14.8	00:53:55	35.4
16	8	00:07:43	16.8	00:43:23	39.7
17	8	00:06:51	13.7	00:34:52	32.1
18	4	00:19:27	15.4	00:57:40	55.2
19	4	00:12:03	20.4	00:42:20	42.1
20	4	00:13:25	24.6	00:49:11	51.2
21	2	00:15:43	25.4	00:52:52	52.2
22	2	00:17:01	28.1	00:55:53	54.9

Table 5.1: Comparison of running time and memory usage. Running time and memory usage for IMPUTE neighbour select algorithm (IMPUTE5 NS) and IMPUTE5 using all the reference panel states (IMPUTE5 ALL STATES) for imputing all the 2504 samples from 1000 Genome Project using 1000 Genomes Project itself as a reference panel. The results were obtained by running each chromosome at the time with all the chunks in parallel. Chromosomes divided in the same number of chunks have different running time due to the length of chromosome. Running time and memory usage are affected by the number of chunks used and the length of each imputed region.

We then extended this to populations using 1000GP data. We found bigger values of the expected sequence sharing between members of the same population and members of the same super-population, and these results agreed for both the IMPUTE5 settings used.

The idea of using the same dataset as in a reference-target framework to cluster haplotypes and running a state selection algorithm can potentially be adopted by other methods, for example in the context of IBD detection methods.

We also used an imputation testing framework for WGS dataset made available in the IMPUTE5 software. By using the original WGS dataset as a reference panel, and the dataset, but masked at non-chip sites, as a target panel, we run imputation by banning from the copying states the reference haplotypes corresponding to the target. We used this to impute the 1000 Genome Project dataset using all the reference panel states and the neighbouring selection algorithm. We again found good concordance in terms of accuracy between the two settings. The leave-one-out testing framework can be used in the context of genotype imputation in many ways, for example to verify the imputation accuracy of several SNP arrays.

6

A Pipeline for Genotype Imputation

Contents

In this chapter, we discuss of a pipeline for genotype imputation which uses IMPUTE5, designed to impute from a reference panel of phased haplotypes into a genome-wide association dataset. We see how quality control (QC) techniques can be applied to both the GWAS dataset and to reference panel in order to obtain a good quality imputation dataset.

We show the preferred strategy to use IMPUTE5 in a genome-wide analysis, which involves pre-phasing as a preliminary step of imputation, and considering different non-overlapping chunks of the genome for imputation. The pipeline stitches

the imputation results back together into whole-chromosome output files, which can be checked in a post-imputation assessment.

6.1 Quality control for GWAS

Genome-wide association studies are based on genotyping chips that allow researchers to interrogate hundreds of thousands of SNPs across the entire genome. A crucial step, which is part of any GWAS, is the use of Quality Control (QC), resulting in a reduction in both the number of individuals and number of markers used for the analysis. The use of QC filters is necessary in order to generate reliable results and remove low quality data, for example to remove poorly genotyped SNPs (Coleman et al. 2016) due to genotyping error or uncertainty in the genotype calling. In the context of GWAS, a small amount of error can propagate for the whole downstream analysis. The increased sample size of the datasets currently available, such as the UK Biobank, can amplify false positive signals if QC is not well performed (Turner et al. 2011).

To appropriately perform quality control to a GWAS dataset, a set of filters is applied. These filters can be logically divided in a sample level QC and a marker level QC.

The sample level QC involves examining:

- genotype call quality and missing rates;
- sex discrepancy, verifying that there is correspondence between sex chromosome genotypes and reported gender;
- heterozygosity, filtering individuals having high or low heterozygous rates;
- relatedness between genotyped individuals, by calculating IBD of all sample pairs;
- population stratification, clustering samples using IBS and filter the outliers.

The marker level QC involves examining:

- call rates, removing low genotype calls;
- minor allele frequencies, where low values mean that there is lower power in detecting associations and sites are more prone to genotyping errors, as genotype call algorithms have less information.
- deviations from Hardy-Weinberg equilibrium, a commonly used indicator of genotyping error.

Therefore, QC filter a dataset to a subset of very confident markers and samples. Different protocols are available and the specific QC filters and thresholds to apply to a GWAS dataset depend on the population, accuracy of the genotype calls, and other factors. For a pipeline describing in details all these aspects there are several tutorial available, e.g. (Turner et al. 2011; Ellingson and Fardo 2016; Marees et al. 2018).

When imputation on a GWAS dataset is performed, data is assumed to have already been filtered. Additional QC steps are needed before the imputation in order to check the consistency of reference panel data and the GWAS dataset. For these reasons, QC is applied to the reference panel, target panel and the imputed dataset removing deviations from the expected distributions of MAF.

6.2 Methods

6.2.1 Pipeline

We developed a genotype imputation pipeline to impute from a reference panel of phased haplotypes into a genome-wide association dataset. It uses SHAPEIT4 for pre-phasing and IMPUTE5 for imputation. The pipeline is available at: `https://github.com/SimoneRubinacci/IMPUTE5_1000G_PIPELINE` and is based on the publicly available FIMM imputation pipeline (Pärn et al. 2019).

Software In order to run the pipeline we use a set of standard programs to perform QC and to analyse the data. The software package required from the pipeline are indicated in Table 6.1.

Software	Version	Function	Required
bcftools	>=1.7	VCF files and QC	Yes
samtools	>=1.7	FASTA reference files	Yes
R	3.4.1	Data analysis / plotting	No
R package data.table	-	Data analysis / plotting	No
R package sm	-	Data analysis / plotting	No
SHAPEIT4	>=1.0	Haplotype phasing	Yes
IMPUTE5	>=1.0	Genotype imputation	Yes
imp5Converter	>=1.0	File format for reference panels	No

Table 6.1: Software packages used by the imputation pipeline.

Non-mandatory software packages are strongly recommended for a smooth usage of the pipeline. For example, part of the QC is performed using the R software. Although the pipeline can work without applying all the QC steps, it is strongly recommended to run all the steps.

PATH variable First of all, it is necessary to export the paths of the software location into the variable PATH. In addition, it is also necessary to export the variable BCFTOOLS_PLUGINS, since the pipeline uses the bcftools fill-tags plug-in to add information in the INFO field of VCF files.

Pipeline overview The pipeline is composed of 5 steps:

1. Downloading the 1000 Genomes project (if necessary) and basic QC on the reference panel. As a QC step, it performs the alignment with the reference genome build 38. Binary representation of the reference panel specifically for IMPUTE5 (IMP5) is also created;

2. Basic QC of the target dataset. Starting from a set of target genotypes, QC analysis is performed on the target markers, including consistency checks with the reference panel. This step creates a set of genotypes on which we can do pre-phasing;

3. Pre-phasing using SHAPEIT4 on the set of genotypes obtained in the previous step. To increase accuracy of the phase, we also use the reference panel obtained during the first step;

4. Imputation using IMPUTE5 on the phased target haplotypes obtained in the previous step. Imputation always requires a reference panel of haplotypes, and we use the dataset obtained in the first step;

5. Post-imputation analysis of the generated imputed dataset.

In addition to this, a preliminary step called 'step 0' is part of the pipeline, settings up the necessary files in order to prepare the environment.

6.2.2 Step 1: reference panel

A reference panel of haplotypes is needed for imputation and pre-phasing. For this reason, if a reference panel of haplotypes is not available, step 1 of the pipeline downloads the 1000 Genomes Project phased data on build 38 as a reference panel. QC is then performed on the reference panel.

Download reference sequence for alignment As a first step the script downloads the reference sequence for alignment, then it downloads 1000 Genomes phased data, and creates a temporary file used for the QC step.

```
# 1. Download reference sequence for alignment and index
wget -O- ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/
   ↪ GCA_000001405.15_GRCh38/seqs_for_alignment_pipelines.
   ↪ ucsc_ids/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna.gz
   ↪ | \gzip -d > ${REF_DIR}/HumanAssembly.fna
samtools faidx ${REF_DIR}/HumanAssembly.fna
```

Download 1000 Genome Project Phased 1000 Genome Project dataset is publicly available and can be used a reference panel.

```
# 2. Download 1000 Genomes phased data
wget http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/
   ↪ supporting/GRCh38_positions/ALL.chr{{1..22},X}_GRCh38.
   ↪ genotypes.20170504.vcf.gz{,.tbi} -P ${REF_DIR}
```

```
for CHROM in {1..22} X; do
  mkdir -p ${REF_DIR}/chr${CHROM}
  mv -f ALL.chr${CHROM}_GRCh38.genotypes.20170504.vcf.gz* ${
     ↪ REF_DIR}/chr${CHROM}
done
```

Basic quality control QC on the reference dataset performs the following steps, in order:

- Rename chromosomes with the 'chr' tag (*bcftools annotate –rename-chrs*). A pre-formatted file is created for this before applying QC;
- Remove singletons (*bcftools view -e*);
- Split multi-allelic sites into multiple rows so that each site can be considered biallelic (*bcftools norm -m*);
- Keep only SNPs and indels (*bcftools view -i*);
- Align the variants to reference genome;
- Annotate the file with unique marker IDs;
- Remove sites containing missing genotypes (*bcftools view -g*)

Variant normalisation is also required. The goal is to have a unique representation of the variant shared between reference panels and target datasets. Tools like bcftools can perfom variant normalisation and this is a key step, because mismatch in variant representation will frequently result in inaccurate analyses.

```
# 3. Create a file for mapping chromosome names
:> ${REF_DIR}/chr_name_map.txt
for CHROM in {1..22} X; do
  echo ${CHROM} chr${CHROM}
done >> ${REF_DIR}/chr_name_map.txt

# 4. Perform QC
for CHROM in {1..22} X; do
 REF_FILENAME=ALL.chr${CHROM}_GRCh38.genotypes.20170504.vcf.gz
 bcftools annotate --rename-chrs ${REF_DIR}/chr_name_map.txt \
        ${REF_DIR}/chr${CHROM}/${REF_FILENAME} -Ou | \
 bcftools view -e 'INFO/AC<2 | INFO/AN-INFO/AC<2' -Ou | \
```

```
 bcftools norm -m -any -Ou | \
 bcftools view -i 'INFO/VT="SNP" | INFO/VT="INDEL"' -Ou | \
 bcftools norm -f ${REF_DIR}/HumanAssembly.fna -d none -Ou | \
 bcftools annotate --set-id '%CHROM\_%POS\_%REF\_%ALT' -Ou | \
 bcftools view --no-version -g ^miss -Ob \
        -o ${REF_DIR}/chr${CHROM}/${REF_NAME}_chr${CHROM}.bcf && \
 rm -f ${REF_DIR}/chr${CHROM}/${REF_FILENAME}*
done
```

Other QC steps (not shown) include discarding duplicated samples in the panel and convert chromosome X haploid data to homozygous diploid. Additionally tags of the in the INFO field are added using *bcftools +fill-tags* extension. Finally, when the reference panel is ready, a binary IMP5 file format (one for each chromosome) can be created in order to be used for imputation.

```
## 10. Convert bcf files to imp5 file format.
for CHROM in {1..22} X; do
  imp5Converter \
  --h ${REF_DIR}/chr${CHROM}/${REF_NAME}_tagged_chr${CHROM}.bcf \
  --r chr${CHROM} \
  --o ${REF_DIR}/chr${CHROM}/reference_${REF_NAME}_chr${CHROM}.
     ↪ imp5
done
```

6.2.3 Step 2: GWAS dataset

The target dataset represents a GWAS study genotyped on a SNP array[20]. There are several standard QC steps to perform on the target panel as part of a GWAS as briefly discussed at the beginning of this chapter. Here the QC is focused on the quality of imputation results.

Basic quality control Quality control steps are similar to those applied for the reference panel. Alignment to the human reference genome is one of the most important steps, in order to correct the dataset-specific flips in the allele order.

As performed for the reference panel, we keep only biallelic sites, by splitting multiallelic sites. It is also important to remove any duplicate variants in the dataset that might be present in some SNP array. Excluding ultra-rare variants is also

[20]In this chapter we focus on datasets for GWAS, because this represent the most common scenario. However, the pipeline can be extended to any other scenario involving imputation.

a common quality control step, especially for chip-based data, because genotype calling algorithms tend to have poor quality for low values of minor allele frequencies.

Allele frequency concordance A final quality control step is to check the consistency of the reference and the target allele frequencies at shared variants. This is especially important for big GWAS datasets. In this case, assuming that the reference panel and the study dataset come from the same population, both the datasets should show approximately the same allele frequency at the same variant. For all the shared variants, an easy way to visualise allele frequencies is to plot reference panel AF on the x-axis and target AF is on the y-axis. This should approximately show a diagonal line of points with slope 1. Markers with variation of allele frequency are placed far from the diagonal line. A filtering criteria can be set by excluding variants that are too far from the diagonal point.

A simple script to check allele frequency concordance has been provided in (Pärn et al. 2019). This produces a list of variants that are not concordant and can be excluded from the target dataset.

6.2.4 Step 3: pre-phasing

Pre-phasing is the process of inferring the underlying haplotypes from GWAS genotypes prior to genotype imputation (B. Howie et al. 2012). Pre-phasing allows genotype imputation methods to perform only haploid imputation which is much faster than diploid imputation. In this pipeline pre-phasing is performed using SHAPEIT4 (Delaneau et al. 2018) and could be performed only using the target dataset divided by chromosomes. In order to increase accuracy of the haplotype phase, a reference panel haplotypes is used, particularly the quality-controlled version generated in step 1 of the pipeline.

Phased data of different chromosomes are then concatenated in a unique file. This phased GWAS dataset is the target panel of haplotypes used for genotype imputation.

```
for CHROM in {1..22} X; do
  mkdir -p ${PHASE_DIR}/chr${CHROM}
  ${BIN_DIR}/shapeit4 \
  -H ${REF_DIR}/chr${CHROM}/${REF_NAME}_tagged_chr${CHROM}.bcf \
  -I ${TARG_DIR}/target_all_masked.bcf \
  -M ${MAP_DIR}/chr${CHROM}.b38.gmap.gz \
  -R chr${CHROM} \
  -T 16 \
  --log ${PHASE_DIR}/chr${CHROM}/phased.chr${CHROM}.log \
  -O ${PHASE_DIR}/chr${CHROM}/phased.chr${CHROM}.bcf
  str_concat="${str_concat} ${PHASE_DIR}/chr${CHROM}/phased.chr${
     ↪ CHROM}.bcf"
done

bcftools concat ${str_concat} \
        -Ob -o ${PHASE_DIR}/target.phased.bcf
bcftools index -f ${PHASE_DIR}/target.phased.bcf
```

6.2.5 Step 4: imputation

Genotype imputation runs independently on each chromosome. IMPUTE5 takes advantage of index files to read a region of the reference panel and the target panel quickly in memory. Since imputation runs independently on different chunks of the same chromosome, in order to attenuate the effect of the chunk borders, an extended region on both sides of the reference and target panel is actually considered for imputation. These extended regions are called imputation buffers.

Imputation quality is dependent on the size of the chunks and the size of imputation buffers. A typical size used for imputation is 5 Mb (B. N. Howie et al. 2009; Bycroft et al. 2018), a reasonable trade-off between accuracy and memory requirements. With an increment of the window size to 10 Mb or 15 Mb should be associated a small increase in accuracy (B. L. Browning and S. R. Browning 2016). In addition to this, a good idea would be to choose window size based on the genomic distance instead of physical distance.

Several methods can be used to determine the size of the window. IMPUTE5 accepts any genomic region in the standard interval used by bcftools and SHAPEIT4, *chr:from-to*, where *from* and *to* are two positions on the chromosome *chr*. In the pipeline, chunks are provided in the form of a text file, one for each chromosome.

Chunks are approximately of size 10 Mb. The buffer parameter used is 1 Mb for each imputation task.

```
for CHROM in {1..22} X; do
  mkdir -p ${IMPUTE_DIR}/chr${CHROM}
  mkdir -p ${IMPUTE_DIR}/chr${CHROM}/log
  mkdir -p ${IMPUTE_DIR}/chr${CHROM}/bcf
  input=${IMPUTE_DIR}/chr${CHROM}/chunks.txt
  i=0
  str_concat=""
  while IFS="   ", read -r reg_buf reg_no_buf remainder
  do
    printf -v j "%02d" $i
    ${BIN_DIR}/impute5 \
    --h ${REF_DIR}/chr${CHROM}/reference_${REF_NAME}_chr${CHROM}.
       ↪ imp5 \
    --g ${PHASE_DIR}/target.phased.bcf \
    --m ${MAP_DIR}/chr${CHROM}.b38.gmap.gz \
    --r ${reg_no_buf} \
    --ne 20000 \
    --b 1000 \
    --pbwt-depth 8 \
    --pbwt-cm 0.1 \
    --thread 1 \
    --log ${IMPUTE_DIR}/chr${CHROM}/log/target_imputed.chr${CHROM
       ↪ }.${j}.log \
    --o ${IMPUTE_DIR}/chr${CHROM}/bcf/target_imputed.chr${CHROM}.$
       ↪ {j}.bcf &

    str_concat="${str_concat} ${IMPUTE_DIR}/chr${CHROM}/bcf/
       ↪ target_imputed.chr${CHROM}.${j}.bcf"
    i=$((i + 1))
    if ! ((i % 16)); then
      wait
    fi
  done < $input
  wait
  bcftools concat ${str_concat} -n -Ob \
    -o ${IMPUTE_DIR}/chr${CHROM}/target_imputed.chr${CHROM}.bcf
done
```

6.2.6 Step 5: post-imputation quality control

Imputation produces one file per chromosome. One useful metric to check the quality of imputation is the info score. It is possible to compute the info score at each marker using bcftools together with other useful metrics like the p-value of the HWE.

Again, (Pärn et al. 2019), provided a script to plot the distribution of info-scores and concordance of allele frequencies between the imputed dataset and

the reference panel. This can give us a visual idea of the quality of imputation for our GWAS dataset.

An info-score of α in a sample of N individuals indicates that the amount of data at the imputed SNP is approximately equivalent to a set of observed genotypes in a sample of size αN. A typical filter used in GWAS is to discard variants having info-score lower than 0.3-0.5. The actual value used is also dependent on the number of individuals present in the GWAS, where higher values of N implies that a smaller value of the info-score threshold can be used.

6.3 Results

In order to test and validate the pipeline, we run step 1 to 5 on data from the 1000 Genome Project. We first downloaded the reference panel following step 1 of the pipeline. Then we extracted 52 samples from 26 different populations as performed in Chapter 3 section 3.3.1. We used the 1000 Genome Project dataset, except for these 52 samples, as the reference panel. The 52 samples were utilised as a target panel. We masked positions not present in the Infinium OmniExpress-24 v1.2 array.

The testing pipeline described in this section is available here: `https://github.com/SimoneRubinacci/IMPUTE5_1000G_PIPELINE/blob/master/run-example.sh`.

6.3.1 Allele frequency concordance

The GWAS dataset is composed of 52 samples, meaning that the sample size is quite small. In addition to that, both the reference panel and the GWAS dataset contain data from different populations. This means we should not expect good concordance between the reference panel and the study samples. For this reason, exclusion lists for concordance variants is not used.

A picture of the concordance between variants across chromosomes 1-22 and X of the GWAS dataset and the reference panel is given in Figure 6.1. The picture shows there is discordance in allele frequency between the GWAS dataset and the reference panel. The red dots in Figure 6.1 (left) are markers where allele

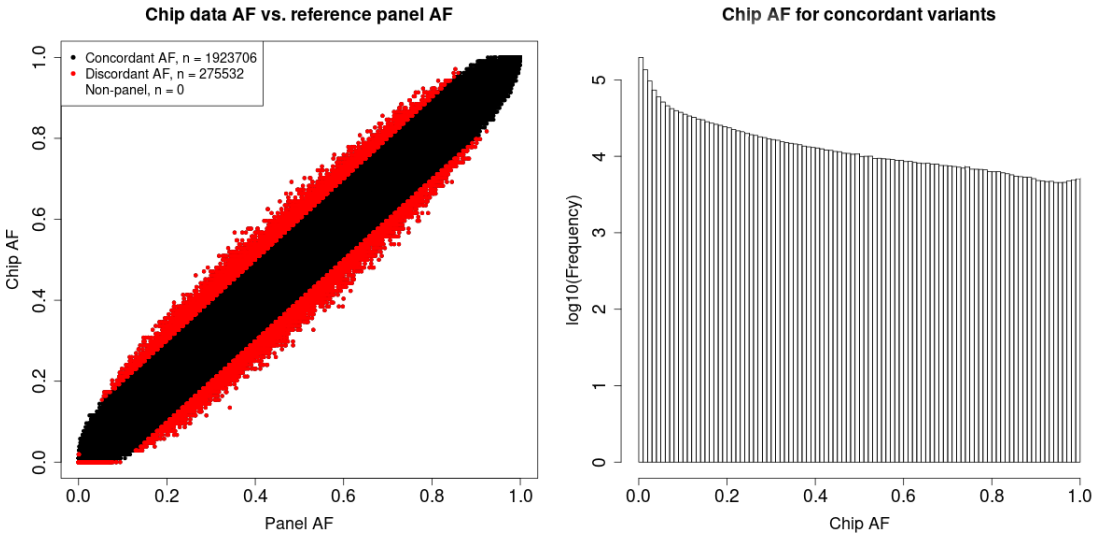

Figure 6.1: Allele frequency concordance. Concordance between GWAS and reference panel AFs at chip markers for chromosomes 1 to 22 and X (left), and distribution of allele frequencies for concordant variants (right). In the left figure, concordant variants are defined as the variants in which the difference in AF is less than 0.1%, or the change fold is between -5 and 5. Discordant variants are shown in red. The histogram on the right shows the frequency for each AF value in the target dataset for concordant variants only. The y-axis is in log scale. As expected, it is an approximately smooth histogram, with a peak for small allele frequencies. The figure has been plotted using script provided in Pärn et al. 2019.

frequency difference between the panels is greater than 0.1% or the fold change between them is, in absolute value, greater than 5.

For the test analysis, the filter by AF discordance has not been applied. However, this is performed in the original pipeline.

6.3.2 Imputation

We re-phased the 52 target samples with the 1000 Genome Project reference panel using SHAPEIT4. The time spent to phase the samples on the 23 chromosomes was approximately 46 minutes using 16 threads on a 16-core computer with Intel Xeon CPU E5-2667 3.20GHz processor.

Imputation was performed in chunks of size 10Mb. The total time spent for imputation is approximately 6 minutes for the 23 chromosomes using a maximum of 16 parallel imputation jobs at a time.

Info-scores were generated using bcftools. Figure 6.2 (top-left) shows the density distribution of info scores of imputed data for the groups of markers having imputed MAF $> 5\%$, imputed MAF between 0.5% and imputed 5% and MAF $< 0.5\%$ for chromosome 20. We can see that the group with MAF $> 5\%$ contains mainly info-values in the range 0.8-1 and, in general, the groups having MAF $> 0.5\%$ have associated values of info-score $> 0.5\%$.

Concordance of allele frequency for imputed data and the reference panel is shown in Figure 6.2. The overall shape is rather wide due to the small size of our target panel and the different populations included in the study. However, we can see that there is a similarity in the AF concordance between imputed data and the reference panel and markers only present in the SNP array and the reference panel, shown in Figure 6.1.

Figure 6.2 shows the histogram of AF distribution of imputed variants and concordance between imputed markers and reference panel along the chromosome. The overall shape of the frequency distribution of imputed variants is as expected, where the majority of the imputed markers have small MAF. The AF differences along the chromosome show that the wide shape in the plot is consistent in all the chromosome and probably due to the small size of the target panel.

6.4 Summary and discussion

In this chapter, we have presented an imputation pipeline that does quality control on both the reference and target panel, performs pre-phasing and imputation, and finally produces qualitative plots that summarise imputation quality.

The pipeline presented in this chapter can help researchers to run phasing and imputation on their GWAS samples. The flexibility of the pipeline consists of:

- using a population-specific reference panel or download 1000 Genome Project (script 1);
- using a custom GWAS dataset or extracting a test dataset from 1000 Genome Project (run-example script).

- running pre-phasing on the whole genome using a reference panel of haplotypes (script 3).
- running imputation on the whole genome (script 4). The provided chunks for imputation can be changed and adopted to user's needs.
- plotting report summaries of the data as part of the pipeline.

This pipeline could be easily adapted to be used on an imputation server. The advantage would be to provide an interface that allows researchers who do not have access to reference panels to use the resource to phase and impute their own samples.

Freely available Software As A Service (SaaS) platforms such as CloudGene (Schönherr et al. 2012), provide a graphical user interface and encrypted layers for cloud-computing pipeline services specifically designed for bioinformatics and in particular genotype imputation. The Michigan imputation server (Das et al. 2016) is based on a set of scripts run inside the CloudGene framework[21]. CloudGene supports the workflow needed for this pipeline, including data transfer, pipeline execution and data export, and represents a well-suited environment for every imputation server.

[21] `https://github.com/genepi/imputationserver`. Source code for the Mitchigan imputation server is available at: `https://github.com/genepi/imputationserver`.

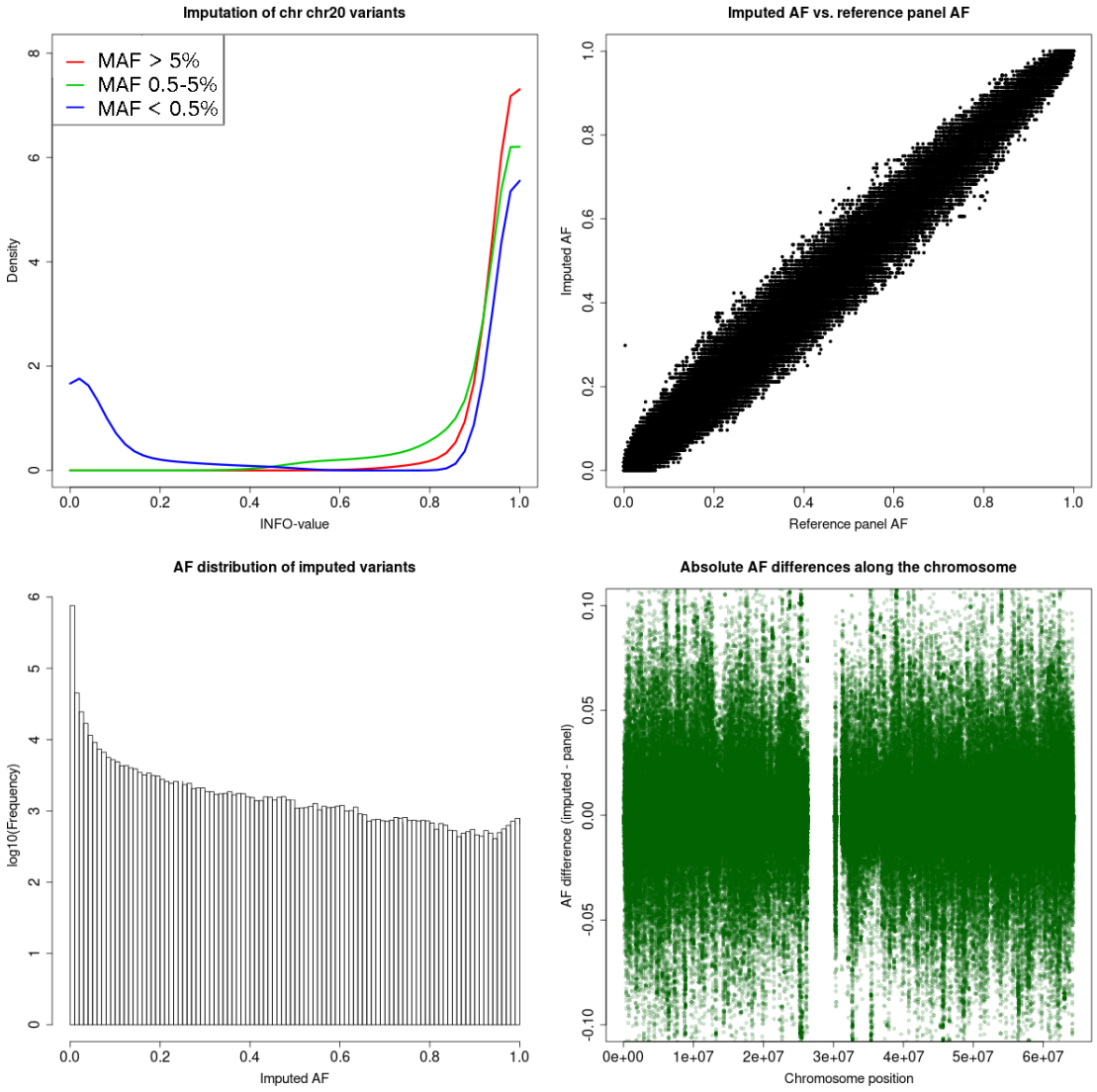

Figure 6.2: Post-imputation quality analysis for chromosome 20. Info-score density distribution of three different AF groups (top-left). Concordance between imputed and reference panel AFs (top-right). As expected, we see a very similar distribution to Figure 6.1, meaning that the distribution of imputed allele frequencies is similar to the distribution of alleles at chip sites.
Allele frequency distribution of imputed variants (bottom-left). The y axis is on a log scale. This distribution is more skewed than the distribution shown in Figure 6.1. This is expected, as most of the imputed variants are rare. Distribution of the difference of AFs between the imputed dataset and the reference panel along the chromosome (bottom-right). The white area at the centre of the plot represents the centromere.
The figure has been plotted using script provided in Pärn et al. 2019.

7

100,000 Genomes Project Imputation

Contents

7.1 Introduction

The 100,000 Genomes project was announced in 2012, and aimed to sequence a total of 100,000 individuals between patients diagnosed with rare diseases or cancer and their unaffected family members (Turnbull et al. 2018). The project is maintained by Genomics England, a private company formed by the Department of Health ans Social Care of the English government. For this reason, it is common to refer to the 100,000 Genomes Project also as GEL. Genomics England works in partnership with the British National Health Service (NHS) to integrate WGS data into the NHS systems.

Sequenced genomes collected from the 100,000 Genomes Project are used for scientific research purposes or can benefit directly patients, by helping the diagnostic process and personalised care. In addition to sequence data, patient records are also collected, providing a rich resource of genomic medical information. In many respects, the project is one of the most important schemes of this type in the world.

The data is accessible for research purposes through a coalition of NHS and academic researchers that form the GE Clinical Interpretation Partnership (GeCIP), which itself is composed by 42 domains based on specific research fields. In terms of population and statistical genetics, data have been made accessible only in an anonymous format.

The data has been collected to have a wide range of phenotypes. Patients have been recruited and grouped based on different disorders within the rare disease cohort. A big effort has been spent in recruiting trios of individuals and other family members. The use of trios is particularly useful in disease genetics, allowing effective variant filtration. Sequence data of a trio, consisting of data of a child and both parents, represents the gold standard for haplotype phasing algorithms, and effectively can be used to test the quality of phasing methods.

Another major project is the UK Biobank (Bycroft et al. 2018). The UK Biobank dataset contains genotypes of almost 500,000 participants, genotyped using the UK Biobank Axiom Array and the Applied Biosystems UK BiLEVE Axiom Array by Affimetrix, on approximately 800,000 markers. The UK Biobank Axiom Array was designed to capture a range of genome-wide genetic variation by carefully choosing the markers. The array contains markers that were already known to be in associated with diseases, but also includes coding variants with minor allele frequencies, and markers specifically chosen to help imputation in European populations, for both the common (>5%) and low frequency (1-5%) MAFs.

The resource is open-access and it represents a major resource for GWAS. The number of samples of the UK Biobank dataset makes possible GWAS on extremely rare alleles, increasing the power of the study. Recently, the resource has been imputed using the HRC reference panel (Bycroft et al. 2018). This

increased the number of markers available for genome-wide association studies to approximately 96 millions. The use of the HRC reference panel for imputation of an dataset as UK Biobank, represented a big challenge for imputation methods. The IMPUTE4 method was specifically developed to impute the UK Biobank dataset using the HRC reference panel.

In the last three years, methods for genotype imputation increased the efficiency of genotype imputation of at least one hundred times. With new-generation imputation methods is now possible to impute the UK Biobank with an even bigger and much more dense reference panel, as the 100,000 Genomes Project, thus providing a more accurate dataset containing extremely rare variants. The use of 100,000 Genomes Project to impute the UK Biobank should provide very accurate imputation, due to the strong shared ancestry of the samples between the projects. Therefore, imputation using the 100,000 Genomes Project can potentially undercover new discoveries of the UK Biobank datasets.

In this chapter, we describe the current state of the project regarding the imputation of UK Biobank using 100,000 Genomes Project reference panel. The project is a collaboration of several researchers and is composed of different steps, from the creation of the a reference panel of haplotypes for the 100,000 Genomes Project to genotype imputation of the UK Biobank resource using the IMPUTE5 software. We tested genotype imputation accuracy of the 100,000 Genomes Project reference panel using 1000 Genomes Project data. We used a preliminary version of the reference panel containing data for chromosome 20 in this preliminary study and we aim to expand these results to the whole genome for the first release of the dataset.

7.2 Methods

The project consists of several steps which can be summarised as follows:

1. preparation and quality control of GEL reference panel;

2. preparation and quality control of the UK Biobank resource for imputation (target panel);

3. developing of a genotype imputation pipeline for the UK Biobank resource;

4. testing of the imputation accuracy of the reference panel using the 1000 Genomes Project and comparisons with other reference panels (HRC and 1000 Genomes project).

Step 1, is the work of Sinan Shi, who, under the supervision of Prof. Jonathan Marchini, performed QC on the original sequenced data and haplotype phasing in order to create a reference panel of haplotypes. In the result section we show results of the reference panel chromosome 20, the only phased chromosome available at this stage of the project.

7.2.1 100,000 Genomes Project dataset

Quality control

Sequence data need to be carefully processed. Quality control is a critical step to generate a subset of consistent data relying on a set of markers where genotyping errors and other artifacts have been minimised. The QC pipeline used generate a reference panel from WGS data involved the following steps:

1. identify duos and trios in the dataset;

2. discard bad quality genotypes, by setting as missing those with genotype quality < 15 and genotype depth < 10;

3. remove sites where missing rate is higher that 5%;

4. apply an allele balance filter[22], by calculating the ratio of allele balance for heterozygous calls and excluding those having a value outside the interval 0.25-0.75;

5. compute the Mendelian errors on duos and trios and remove sites if Mendelian error is > 3;

[22]The allelic balance filter attempts to estimate whether the data supporting a variant call fits the expectation, or whether there might be some bias in the data

Most of these steps have been performed with bcftools software. For the Mendelian error filter, the software plink (Chang et al. 2015) was used. At the end of these steps, haplotype phasing can be performed.

Phasing

Information contained in duos and trios can be used to infer accurate phasing. These accurately phased haplotypes, can be exploited in the form of a reference panel, to increase the accuracy of phasing for unrelated samples in a dataset. This is used by the pipeline adopted for the phasing of 100,000 Genomes Project, summarised as follows:

1. exclude singletons not in duos or trios, because phasing algorithms produce random phase for these genotypes[23];

2. phase duos and trios, obtaining approximately perfect phasing for those samples;

3. phase all the unrelated samples, using the phased duos and trios from step 2 as a reference panel;

4. merge the dataset obtained at step 2 with the dataset obtained at step 3.

The phasing process of the dataset was run using SHAPEIT4 and is still in progress. In the meantime we created a preliminary imputation reference panel for chromosome 20. It contains data of 59,349 individuals and 6,860,694 sites.

In order to test the quality of the phasing process, Sinan Shi created a reduced reference panel of 28,000 samples for chromosome 20, using a previous release of the GEL data. He phased 200 trio parents genotypes using the previously described GEL reference panel and the HRC reference panel. Since the datasets rely on a different set of markers, only sites in common have been considered. Quality of the phasing has been assessed using children of the trios.

[23]It is not the case for duos and trios, because family information can help the phasing process.

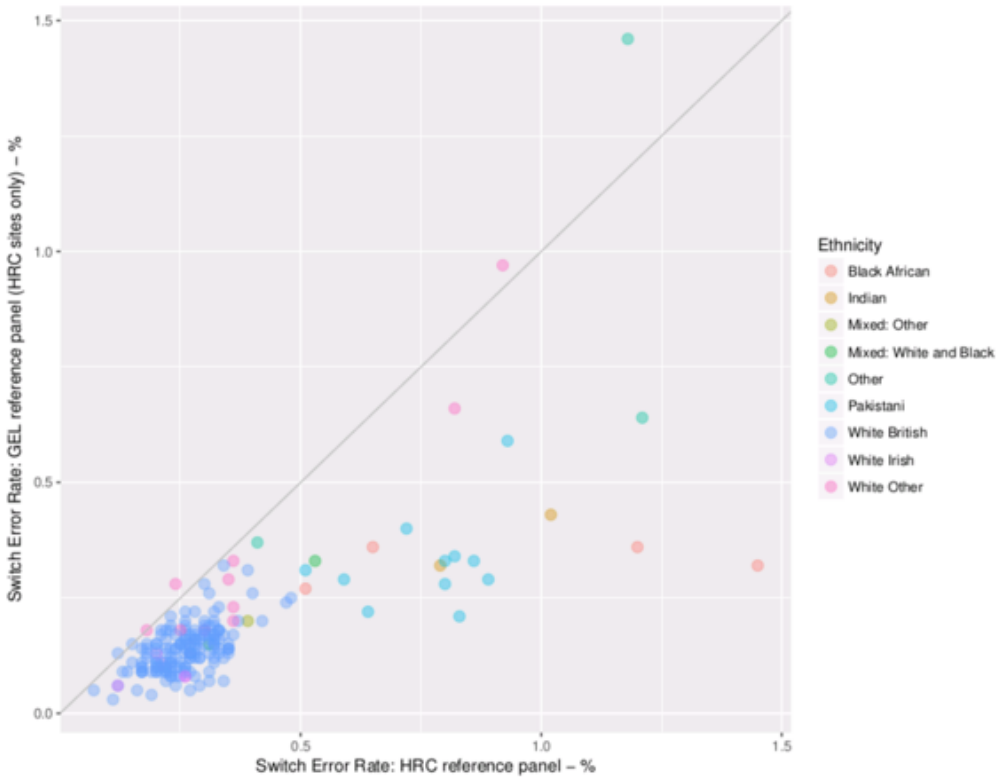

Figure 7.1: Switch error rates of the 200 trio parents phased using the HRC and GEL reference panel. Switch error rates obtained with the HRC reference panel (x-axis) and the reduced GEL reference panel (y-axis). True phased haplotypes has been obtained using trio children information. Almost all the resulting haplotypes obtained a lower switch error rate when the GEL reference panel was used. This figure was prepared by Sinan Shi at the University of Oxford.

Figure 7.1 shows the switch error rate[24] obtained using the HRC reference panel (x-axis) and the reduced GEL reference panel (y-axis). Ethnicity information has been reported on the right side of the plot. It is evident that the switch error rate decreases when the GEL reference panel is used, meaning that the GEL dataset is accurately phased and helps to obtain better phasing for the considered genotypes.

7.2.2 UK Biobank

Release dataset

The UK Biobank release contains genotype and imputed data described in (Bycroft et al. 2018). The release is freely available with prior registration. A description of

[24]The switch error rate is a common measure of phasing quality. A switch error occurs when a site has phase switched relative to that of the previous heterozygous site. The switch error rate is defined as the number of switch errors divided by the number of heterozygous sites minus one.

the dataset is available online[25] or as part of the release, and contains information of the files and quality control filters used for imputation.

We downloaded genotype data of the UK Biobank, applied standard quality control on the data and estimated phased haplotypes (pre-phasing) for chromosomes 1-22. The dataset contains 488,377 samples. We then performed genotype imputation on chromosome 20 data. We applied quality control according the previous imputation study (Bycroft et al. 2018), and we applied a more stringent variant filter.

Variant filtering

As a quality control, we removed markers previously filtered in the previous imputation analysis (Bycroft et al. 2018). This filter removed markers with a missing rate $> 5\%$, and had a minor allele frequency < 0.0001. These markers are identifiable in the public release of the project, using the '*in_Phasing_Input*' field. Applying this filter, we obtained 670,741 markers.

We then applied an additional filter. We firstly computed the Hardy-Weinberg Equilibrium at all variants using white British samples data only. The release of the UK Biobank contains this information in the '*in.white.British.ancestry.subset*' field, obtaining 409,703 white British ancestry samples. We excluded markers in the whole dataset if the Hardy-Weinberg Equilibrium p-value was $< 10^{-12}$ for white British samples as performed in (Bycroft et al. 2018). This reduced the number of markers from 670,741 to 635,086, keeping unchanged the number of samples to 488,377.

Liftover and phasing

We converted the original plink files to vcf file format in order to perform liftover and phasing. Prior to phasing, we mapped the genomic build of the dataset from GRC37 to GRCh38 and aligned to the reference sequence for build 38. We used the software Picard[26] for this. The liftover process was not able to align 504 variants across all the genome, thus resulting in 634,582 variants for the GRCh38 dataset.

[25] `https://biobank.ctsu.ox.ac.uk/crystal/refer.cgi?id=531`
[26] `http://broadinstitute.github.io/picard/`

Haplotype phasing on the auto-somes was carried out using SHAPEIT4.1 (Delaneau et al. 2018). Phasing has been performed using a single job per chromosome.

Phasing quality can be tested using the trios in an external dataset. However, several studies have assessed the quality of UK Biobank phasing obtained using SHAPEIT4 software (Bycroft et al. 2018; Delaneau et al. 2018), and we relied on those results.

Imputation pipeline

Imputation has been performed using IMPUTE5. In order to handle with a target panel of 488,377 samples, we split the phased panel into 49 files containing ~10,000 samples each. In addition to this, imputation was run in chunks of 15 Mb length. This highlights how it is important to read the reference panel efficiently, since the same reference panel is read by several imputation tasks for the same chunk. In our experiments, the time spent reading the reference panel and performing state selection is less than 1% of the total amount of time spent for imputation.

The pipeline presented in chapter 6 can be easily adapted to impute a target panel split on multiple files. At the end of imputation, we merge all the files containing data of a particular chromosome region. This step can be computationally expensive, because it involves decompression on 49 files and compression of the new output file. We use the software qctools for this purpose.

Finally, we append all the files of different regions within a chromosome together, obtaining a single file per chromosome. This latter step is very efficient and does not require compression or decompression.

Computational efficiency

Imputation of the UK Biobank dataset is a computationally demanding process. We split the target panel into 49 subsets containing approximately 10,000 samples and run imputation independently. To impute the whole chromosome 20 we run 196 independent jobs where each one of these was run using 4 threads. The average running time of each job was approximately 50 minutes and each job used approximately 55 GB of RAM. For this reason we limited the number of concurrent

jobs to 10 in order to not drain all the memory of the cluster. The total running time spent for imputation was less than 20 hours to impute UK Biobank chromosome 20.

7.2.3 Genotype imputation accuracy assessment

A natural framework to test the genotype accuracy of different imputation methods is the use of sequence data, where markers are masked during imputation but the true value is known. The 1000 Genomes Project phase 3 is one of the most used datasets for this purpose, since it contains data of individuals across several populations and a fully phased dataset is available from the International Genome Sample Resource website[27].

Data preparation

We used phased data from The 1000 Genomes Project phase 3, released in 2015 (The 1000 Genomes Project Consortium 2015). We removed all the samples not belonging to GBR population and masked variants not present in the Illimina GSA 2 v2.0 A1 array. For the 91 samples in the GBR population, we also kept the fully sequenced dataset for comparison of true allele dosages. This resulted in 91 samples with 1,802,302 markers in the fully sequenced dataset and 14,500 markers in the simulated SNP array for chromosome 20.

In order to perform a comparative analysis we performed imputation of these 91 samples using three different reference panels: (i) 100,000 Genomes Project, (ii) the HRC and (iii) 1000 Genomess Project. For the HRC and 1000 Genomes Project we had to remove the 91 British samples before using them as a reference panel for imputation.

We evaluated imputation performance at the intersection of the three panels, which resulted to have 664,457 markers for chromosome 20, and we used these markers to evaluate imputation performance when comparing datasets.

[27] `https://www.internationalgenome.org/category/phase-3/`

Imputation parameters

We used the IMPUTE5 software for genotype imputation for the three reference panels. As parameters we used a 1 Mb buffer and $L = 8$. Imputation was performed in chunks of 15 Mb in size, resulting in four chunks for the whole chromosome 20.

Genotype imputation accuracy calculation

The imputation accuracy has been estimated as the r^2 correlation between imputed dosages and the true sequence genotypes. For each allele frequency bin, aggregate r^2 was calculated across all sites and all samples.

We binned the r^2 results using the allele frequency from the self-reported white British samples in the 100,000 Genomes Project (n = 37,938).

7.3 Results

7.3.1 Imputation accuracy of GBR population

Figure 7.2 shows the imputation accuracy for the 91 samples of the GBR population, when imputing from GEL, HRC and 1000 Genomes Project reference panel. We estimated the allele frequencies from the self-reported white British samples of GEL dataset. The use of inaccurate allele frequencies could introduce a bias in the r^2 calculations, especially for low allele frequencies.

The figure shoows that the best imputation performance is obtained when using GEL reference panel. The increased accuracy at rare variants, obtained using GEL compared to the HRC, can be explained by the increased number of samples in the reference panel from the same population of the target samples (British population). It is not surprising the drop of imputation accuracy of the 1000 Genomes Project reference panel. This due to the lack of samples with British ancestry in the reference panel and the limited amount of Europeans. Even in this case, this is more evident for rare variants.

Figure 7.2 indicates that, using GEL reference panel for imputation at rare variants (MAF<1%), the r^2 ranges between 0.8 and 0.5 for British samples, a

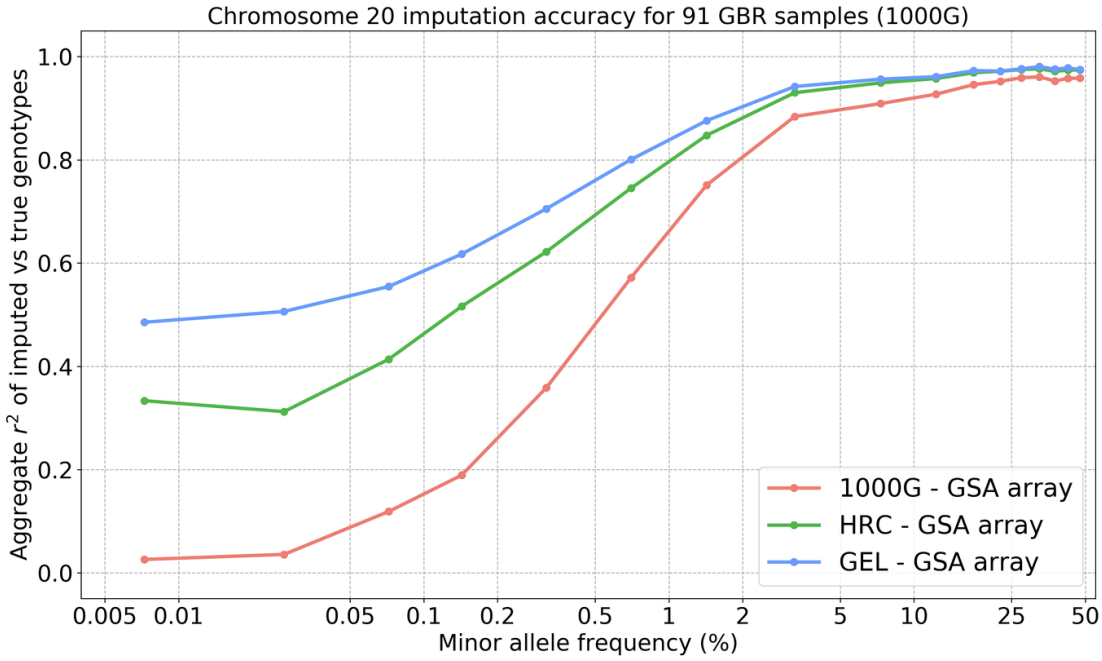

Figure 7.2: Genotype imputation accuracy for GBR population in the 1000 Genomes Project. Aggregate r^2 for 91 GBR samples in the 1000GP dataset when using 100,000 Genomes Project (blue), the HRC (green) and 1000GP (red) reference panel. Allele frequencies have been estimated using self-reported white British information in the GEL reference panel, and used as bins for the r^2 calculations.

result comparable to what found in the TOPMed paper (Taliun et al. 2019) and expected for high-coverage sequencing datasets.

Since the majority of samples in the UK Biobank dataset are from White British ancestry (n=409,724) (Bycroft et al. 2018), the results shown in Figure 7.2 suggest that imputation of UK Biobank using GEL reference panel should be able to get more accurate genotypes compared to imputation from the HRC reference panel, therefore increasing the power of GWAS compared to the data generated in (Bycroft et al. 2018).

Next steps

The results showed in Figure 7.2 are encouraging and suggest that the phasing step of GEL reference panel seems to give reasonable accuracy. However, more validations are required before starting the imputation the UK Biobank dataset. A first step would be to generalise the analysis to the whole genome in order to

validate the phasing of GEL by using genotype imputation. A problem in the phasing would result to badly imputed rare variants. The comparison with the HRC and 1000 Genomes can provide useful hints about problems during the phasing step.

In addition to this, we are planning to use the resequenced (30x coverage) 2504 samples phase three panel from the 1000 Genomes Project by the New York Genome Center (NYGC)[28]. This high-quality dataset can be used for testing and could potentially be included in the GEL reference panel, in order to improve imputation quality for non-European samples.

7.4 Summary and discussion

In this chapter we described an ongoing project aimed at creating a reference panel for the 100,000 Genomes Project dataset and applying it to impute data from the UK Biobank dataset. The project is particularly ambitious as these dataset represents two of the biggest projects currently available in genetics.

We described the process of creating a reference panel for the 100,000 Genomes Project. Sinan Shi created a pipeline performing quality control on sequence data and generate a preliminary preliminary dataset for chromosome 20.

We tested this preliminary version of the reference panel using the 1000 Genomes Project in an imputation experiment. We showed the quality of the imputation using different estimates of the allele frequencies, since the reference panel contains mainly British ancestry. Since estimation of the correct allele frequencies is crucial to obtain accurate r^2 values, especially for low allele frequency variants, demographic inference (e.g. by using Principal Components Analyses) on the reference panel samples could performed, in order to estimate the correct allele frequencies for each population.

We showed how the imputation is performed with the UK Biobank dataset. The process of imputation must be carefully optimised due to the demanding amount of resources required to run imputation with dataset of this size.

In the next few months, a complete version of the reference panel will become available and genotype imputation will be extended to all the chromosomes.

[28] `https://www.internationalgenome.org/data-portal/data-collection/30x-grch38`

The imputation of the UK biobank using the 100,000 Genomes Project resource is extremely valuable. Due to the shared ancestry of the samples of the two datasets, imputation accuracy should reach levels that has never reached before. The increased accuracy and number of rare markers available, makes the UK Biobank dataset the main resource to use for GWAS in the next few years.

8 Conclusions

This thesis explores methods of genotype imputation for reference panels of millions of samples. Reference panel of that size are still not available, but will be in next few years (Bycroft et al. 2018). This new generation of reference panels will increase the cost of imputation for each study sample. Therefore, the development of new methods of genotype imputation is necessary in order to keep the cost of genotype imputation feasible.

In the first chapter, we introduced the problem of genotype imputation in the context of GWAS. We presented the statistical framework and the Li and Stephens model (N. Li and Stephens 2003). We also presented the current state-of-the-art imputation methods and discussed the differences between them.

In chapter 2, we presented the Positional Burrows-Wheeler Transform, a data structure designed to represent efficiently a set of haplotypes in order to allow fast search of sub-sequences. We provided an open-source implementation of the data structure and benchmarked its performance. The software package has been designed as a library, making it particularly suited to be used from other software packages.

Chapter 3 describes the new IMPUTE5 genotype imputation model, which is based on a new state selection metric based on the PBWT. The method proves to be the most efficient genotype imputation method currently available both in terms of memory and computational time. To achieve this level of performance, a

combination of previous advances and new methodological improvements has been used. The development of a new file format, designed to store reference panel data, allows to retrieve a region of the chromosome in constant time, and the specific IMP5 compression does not require a decompression step as, for example, algorithms based on Deflate. The description of IMPUTE5 model is available as a pre-print manuscript[29] and is currently under review in a peer-reviewed journal.

Chapter 4 evaluates the new state selection metric with previously proposed metrics, such as the Hamming distance used by IMPUTE2. We show that our metric is both more efficient and more accurate. In the chapter, we also discuss about a chromosome painting application using the 1000 Genomes Project. We show that selecting a subset of states does not affect imputation accuracy in practice, obtaining very similar results than using the full set of states.

In Chapter 5, we propose a novel way to cluster samples by using the same dataset as reference and target panel, but, when selecting states, we ban the haplotypes in the reference panel that correspond to the target. We show that our state selection has a very similar accuracy as using all the states both in terms of expected amount of sequence copied and genotype imputation.

Since genotype imputation is a crucial step in a GWAS, we provide an easy-to-use pipeline for pre-phasing and genotype imputation in Chapter 6. The pipeline is designed to impute thousands of study individuals from a reference panel that can have millions of samples. We discuss how this pipeline could be easily implemented on an imputation server, which is especially useful for reference panels that are not publicly available.

Finally, in Chapter 7, we discuss the imputation of the UK Biobank dataset using 100,000 Genomes Project as a reference panel. The 100,000 Genomes Project represents the biggest catalogue of British ancestry available and it is composed of approximately 60,000 samples from England. Data has been sequenced at 30x coverage, resulting in six times more markers that the HRC. We developed a pipeline for genotype imputation for this dataset and showed the increased

[29] `https://www.biorxiv.org/content/10.1101/797944v1`

accuracy of genotype imputation when using the 100,000 Genomes Project as a reference panel compared to other reference panels available, such as the HRC or 1000 Genomes Project.

The imputation method described in this dissertation, IMPUTE5, is the most recent of a series of imputation methods started with IMPUTE1 (Marchini, B. Howie et al. 2007). A big advance was realising that the HMM forward-backwards probabilities are typically very sparse and only a subset of them are actually used for imputation; this led to the IMPUTE2 method which also introduced the idea of pre-phasing (B. N. Howie et al. 2009; B. Howie et al. 2012). IMPUTE3 and IMPUTE4 used a more sophisticated representation of the reference panel and have been optimised to perform only haploid imputation (Bycroft et al. 2018). IMPUTE5 extends some of the ideas of IMPUTE2 and IMPUTE4 by selecting a subset of states using the PBWT and representing the reference panel in a similar way as IMPUTE4.

The speed-up obtained by IMPUTE5 allows imputation on a much bigger set of reference haplotypes compared to other imputation methods. Since increasing the size of the reference panel is linked to more accurate imputation, using a method which is able to leverage reference panels with millions of samples increases imputation accuracy compared methods that can only use smaller reference panels.

We also believe that there is space for further improvements. The use of the PBWT for state selection is one feature in common with SHAPEIT4 and a possible extension could be to optimise imputation to be used after the pre-phasing step. When pre-phasing is performed using a reference panel, SHAPEIT4 builds the PBWT of the reference panel and the current phase estimate of the target panel together, in order to provide accurate phasing. Since the same data structure is used in a similar way by the two programs, IMPUTE5's selection algorithm could run as a final step of phasing, or SHAPEIT4 could provide the list of copying states to IMPUTE5 by running IMPUTE5's selection algorithm. Since the PBWT of the reference panel at chip markers has been already built in memory, the mere search of the target haplotypes in the PBWT is only linear in the number of markers and

target haplotypes. This will make imputation completely independent of the number of samples in the reference panel, reducing imputation to a small step of phasing.

It is possible to improve the method by extending the IMP5 file format. As described here, IMP5 format only provides a basic representation of the haplotypes, but additional information can be added related to the PBWT divergence arrays.

The ideas underlying the IMPUTE5 model could also be applied in other research areas, such as imputation of low coverage sequencing datasets. Imputation of low coverage datasets represents a computational challenge because of the probabilistic nature of the input data, under the form of genotype likelihoods. In this setting, current methods for low-coverage imputation rely on diploid models and scale quadratically in the number of samples of the reference panel. This makes imputation with large reference panels prohibitive in practice. We showed that the PBWT can be an efficient and accurate way to subset haplotypes in the reference panel, and this could be directly used by low-coverage sequencing imputation methods to speed-up the diploid imputation model. In addition, a two-step approach can be used by first performing genotype calling on confident genotypes (e.g. covered by a certain amount of reads). Then, a second round of phasing and imputation with SHAPEIT4 and IMPUTE5 can be used to infer the remaining genotypes. In this way, it would be computationally feasible to perform genotype imputation for low coverage sequencing with large reference panels. A similar approach has been suggested in (Homburger et al. 2019).

The cost of whole genome sequencing is decreasing over time. However, genotype imputation still represents a key step in several GWAS in which data collection has been provided using low-coverage or SNP array technologies. To this regard, it is important to note that in specific areas of research, such as in population genetics using ancient DNA, samples are often sequenced with low-coverage sequencing technologies, due to the quality of the sample. In this context, low coverage sequencing followed by imputation can be the only possible scenario. Genotype imputation provides a cost-effective framework in many areas of genomics where whole genome sequencing is technically unfeasible or economically impracticable.

In conclusion, this work proposes and evaluates new methods to use reference panel information in the context of genotype imputation. We also link possible applications of these methods to other research areas of genomic research. With the increasing number of whole-genome sequencing projects that will lead to the formation of new reference panels, we believe that those methods will have an impact in the context of next-generation datasets.

Appendices

A

Additional IMPUTE5 Figures and Tables

A.1 Other figures and tables

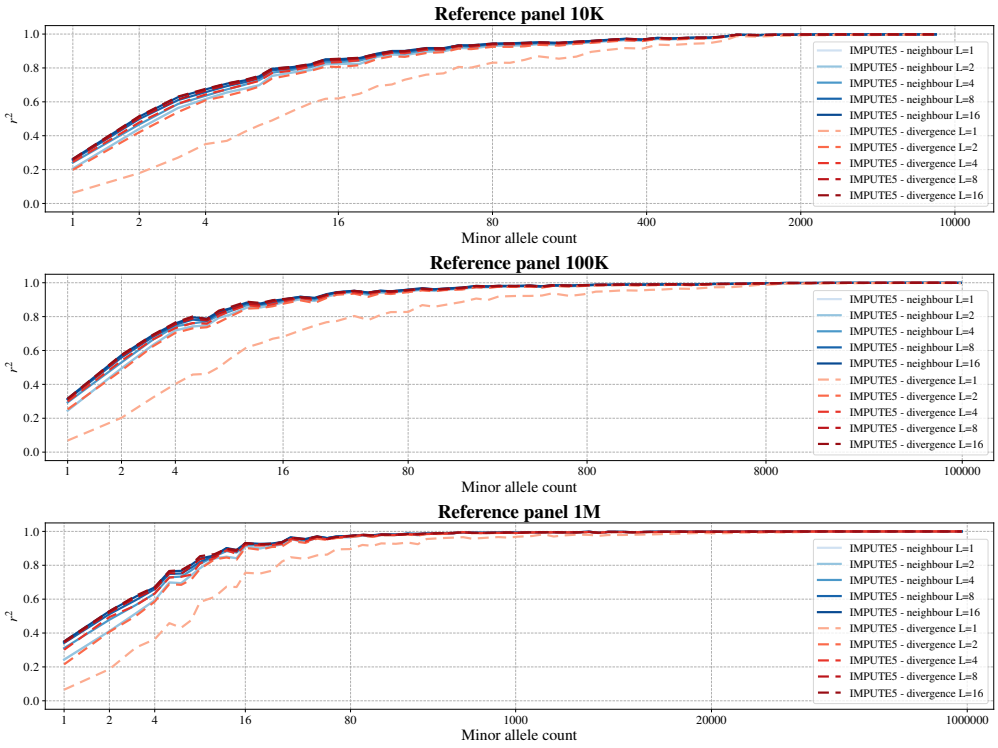

Figure A.1: Imputation accuracy varying parameter L and the selection algorithm using Panel A dataset. Genotype imputation accuracy when imputing genotypes for Panel A dataset for different values of the parameter L using the neighbour selection algorithm and the divergence selection algorithm.

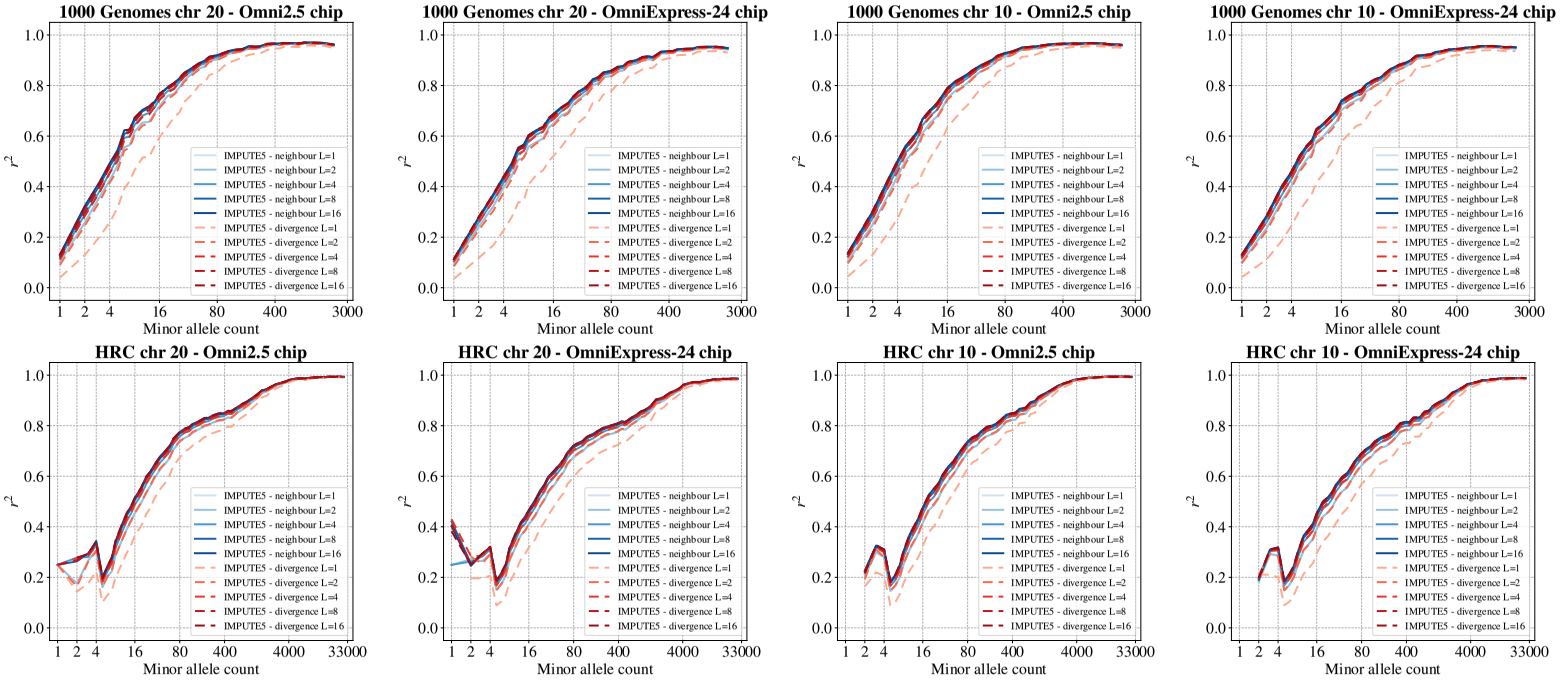

Figure A.2: Imputation accuracy varying parameter L and the selection algorithm using 1000 Genomes and HRC datasets. Genotype imputation accuracy when imputing genotypes using the 1000 Genomes Project reference panel ($n = 2452$) and the Haplotype Reference Consortium reference panel ($n = 31470$) for different values of the parameter L using the neighbour selection algorithm and the divergence selection algorithm.

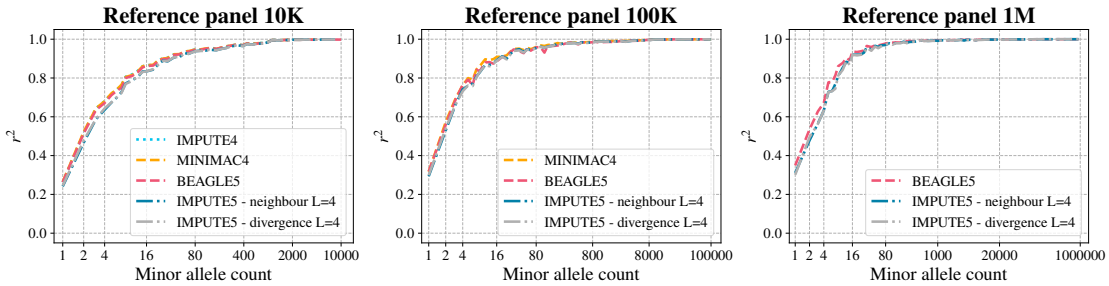

Figure A.3: Imputation accuracy using Panel A dataset using $L = 4$. Genotype imputation accuracy when imputing genotypes from a simulated reference panel of 10K, 100K and 1M UK-European reference samples (Panel A). Imputed alleles are binned according to their minor allele count in each reference panel. The squared correlation (r^2) between the true number of alleles on a haplotype (0 or 1) and the imputed posterior allele probability is reported for each minor allele count bin. The horizontal axis in each panel is on a log scale.

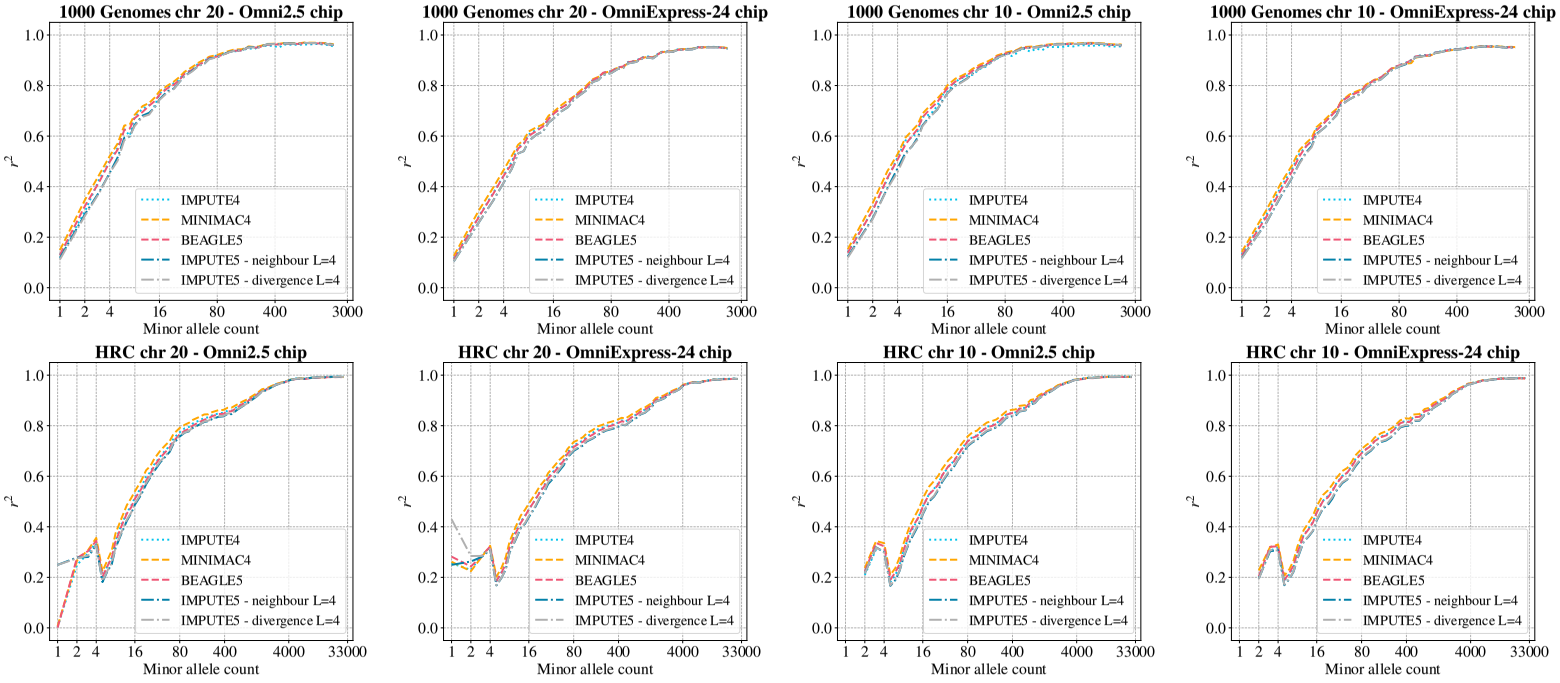

Figure A.4: Imputation accuracy using 1000 Genomes and HRC datasets using $L = 4$. Genotype imputation accuracy when imputing genotypes using the 1000 Genomes Project reference panel ($n = 2452$) and the Haplotype Reference Consortium reference panel ($n = 31470$). Imputed alleles are binned according to their minor allele count in each reference panel. The squared correlation (r^2) between the true number of alleles on a haplotype (0 or 1) and the imputed posterior allele probability is reported for each minor allele count bin.

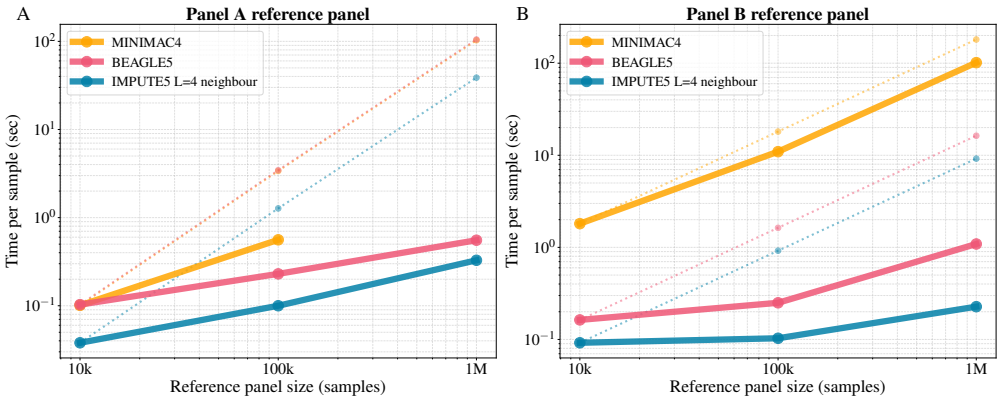

Figure A.5: Per sample imputation time for Panel A and Panel B datasets using $L = 4$. Per-sample CPU time when imputing a 10 Mb region from 10K, 100K and 1M simulated UK-European reference samples into 1,000 target samples using one computational thread. (A) Imputation time when using Panel A dataset (3,333 target markers). (B) Imputation time when using Panel B dataset (33,333 target markers). Hypothetical linear scaling of MINIMAC4, BEAGLE5 and IMPUTE5 are shown as dotted lines, generated by projecting the time using the 10K reference panel. Minimac4 was not able to run using the Panel A 1M reference panel due to time constraints in the construction of the m3vcf file.

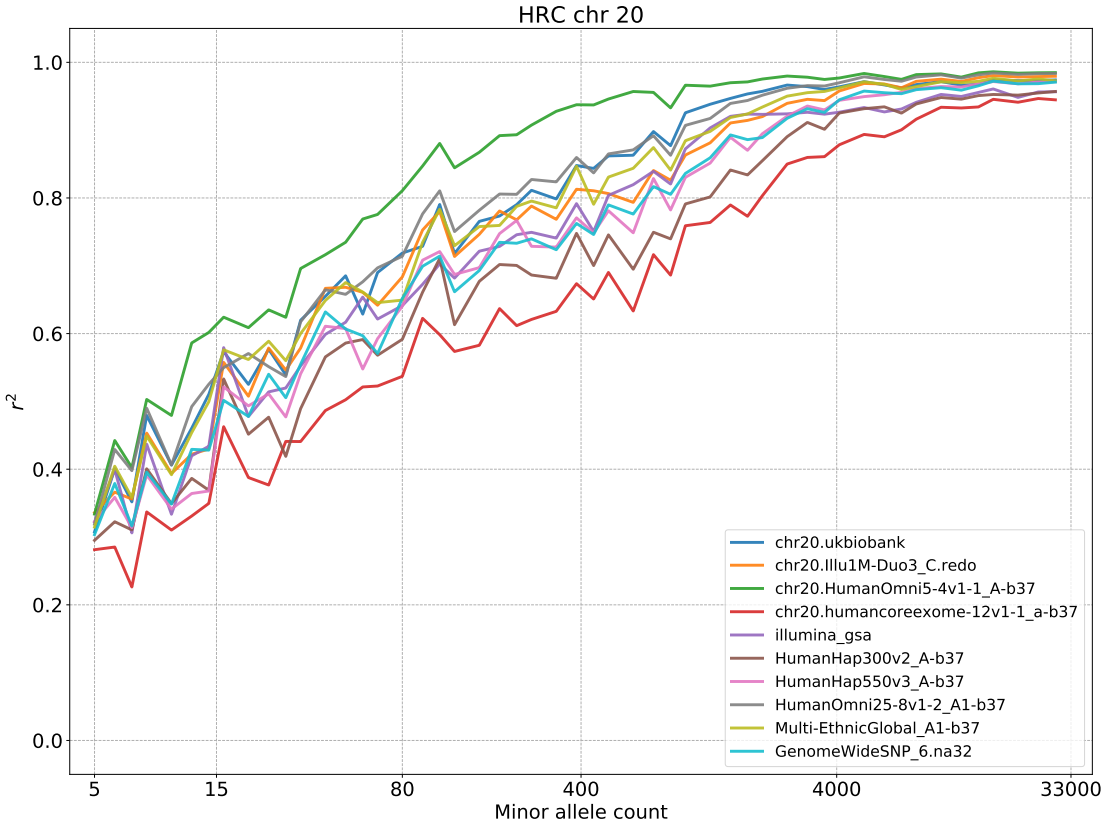

Figure A.6: Imputation accuracy for different SNP arrays using IMPUTE5. Genotype imputation accuracy for chromosome 10 when imputing 10 CEU samples from the 1000 Genomes Project using the HRC reference panel for different SNP arrays using IMPUTE5 neighbour selection algorithm.

Dataset	Single core memory usage (MB)					
	Omni 2.5 chip			OmniExpress-24 chip		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
1000GP chr 20	1,792	7,243	**294**	1,696	6,698	**420**
1000GP chr 10	2,472	11,202	**352**	2,316	11,985	**506**
HRC chr 20	4,842	15,455	**2,015**	4,405	13,361	**809**
HRC chr 10	4,896	17,556	**2,086**	4,417	14,592	**873**

Dataset	Single core time (mm:ss)					
	Omni 2.5 chip			OmniExpress-24 chip		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
1000GP chr 20	02:33	01:24	**00:24**	01:37	01:11	**00:20**
1000GP chr 10	05:16	02:48	**00:50**	03:30	02:31	**00:41**
HRC chr 20	252:59	21:33	**06:58**	124:28	15:26	**04:45**
HRC chr 10	494:59	41:14	**14:26**	250:24	30:16	**10:01**

Dataset	Parallel time (mm:ss)					
	Omni 2.5 chip			OmniExpress-24 chip		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
1000GP chr 20	01:08	00:15	**00:02**	01:08	00:12	**00:02**
1000GP chr 10	02:25	00:28	**00:04**	01:56	00:24	**00:04**
HRC chr 20	27:13	02:12	**00:36**	15:07	01:20	**00:29**
HRC chr 10	56:08	06:07	**01:13**	30:54	02:41	**00:55**

Table A.1: Memory usage and time to impute 1000 Genomes and HRC datasets using $L = 4$. Memory usage and total time to impute a whole chromosome (chromosome 10 and chromosome 20) for 52 target samples when using the 1000 Genomes reference panel and 1,000 target samples when using the HRC reference panel. Time is shown using the format mm:ss. Bold font is used to indicate the method with the lowest time. For the parallel computations, BEAGLE5 and MINIMAC4 were run using a single process with 16 threads; IMPUTE5 was run in 16 (chromosome 20) or 32 (chromosome 10) windows of size ≈ 5 Mb, each of them run in parallel as a different process.

Dataset	**Single core time (mm:ss)**					
	Panel A			Panel B		
	MINIMAC4	BEAGLE5	IMPUTE5	MINIMAC4	BEAGLE5	IMPUTE5
10k	01:41	01:44	**00:39**	08:33	02:43	**02:14**
100k	09:20	03:50	**01:40**	65:15	04:10	**02:22**
1M	-	09:15	**05:29**	1690:50	18:08	**03:47**

Table A.2: Single core time to impute Panel A and Panel B datasets using $L = 4$. Total time to impute $1,000$ target samples in a 10 Mb window using simulation data in Panel A and Panel B dataset. Time is shown using the format mm:ss. Bold font is used to indicate the method with the lowest time. Minimac4 was not able to run using the Panel A 1M reference panel due to time constraints in the construction of the m3vcf file.

B

Additional IMPUTE5 Chromosome Painting Figures

B.1 Other figures

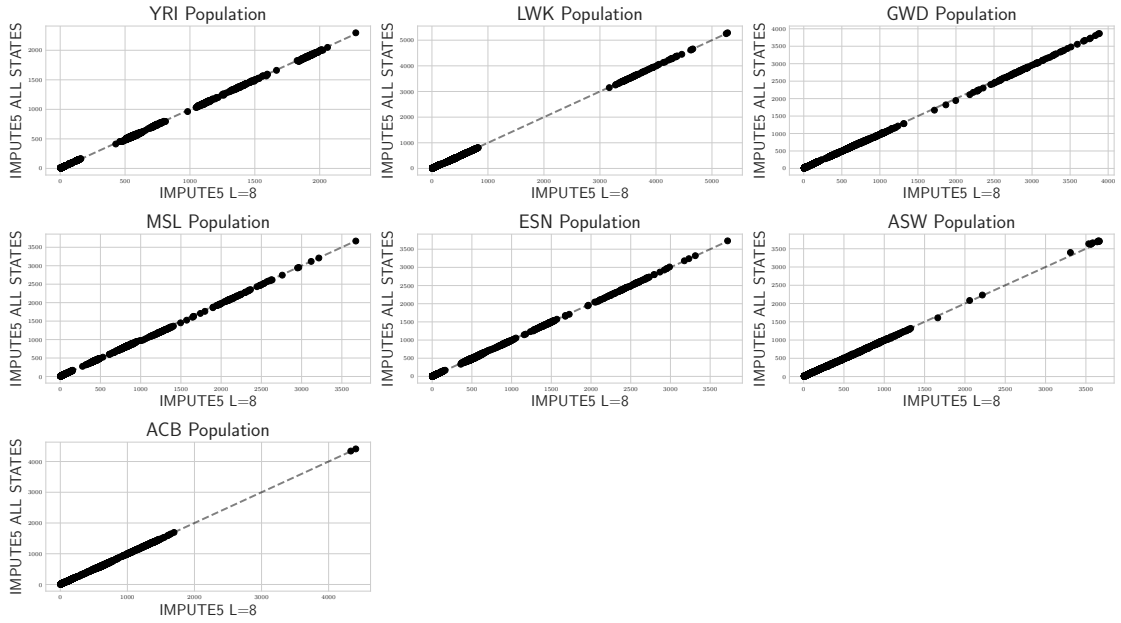

Figure B.1: Concordance between the expected amount of sequence copied from the reference panel (D_i) of IMPUTE5 neighbour select and IMPUTE5 all states for AFR super-population.

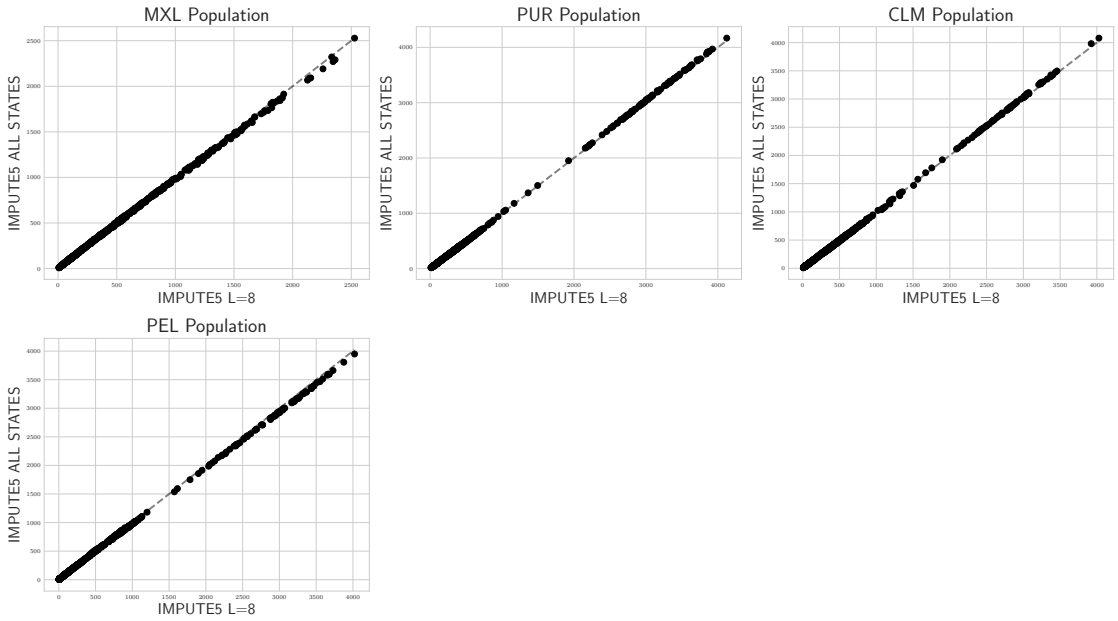

Figure B.2: Concordance between the expected amount of sequence copied from the reference panel (D_i) of IMPUTE5 neighbour select and IMPUTE5 all states for AMR super-population.

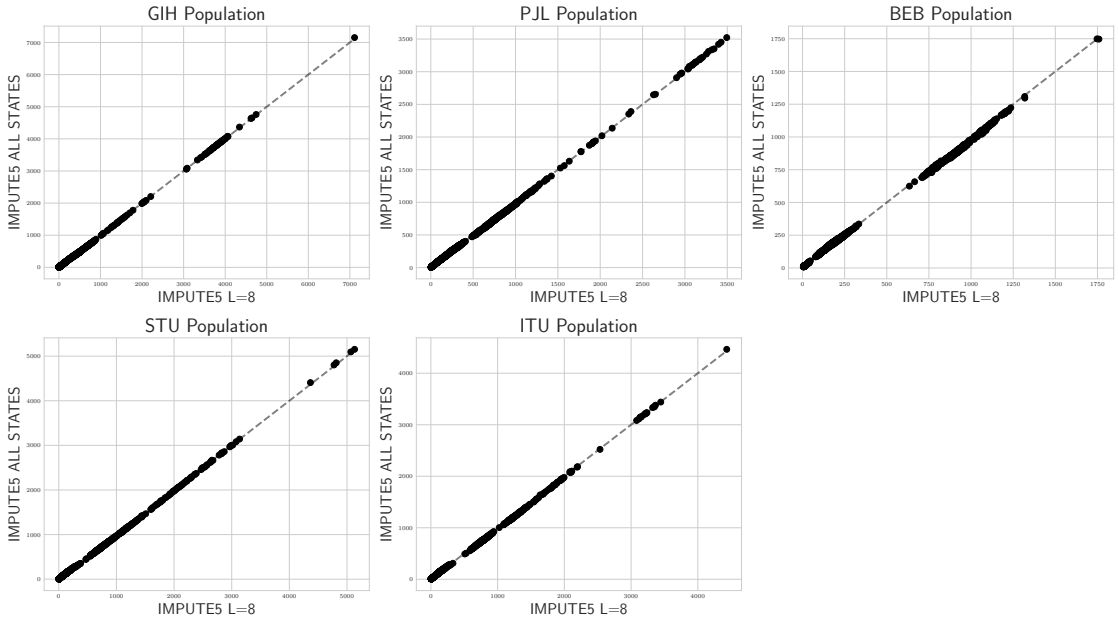

Figure B.3: Concordance between the expected amount of sequence copied from the reference panel (D_i) of IMPUTE5 neighbour select and IMPUTE5 all states for SAS super-population.

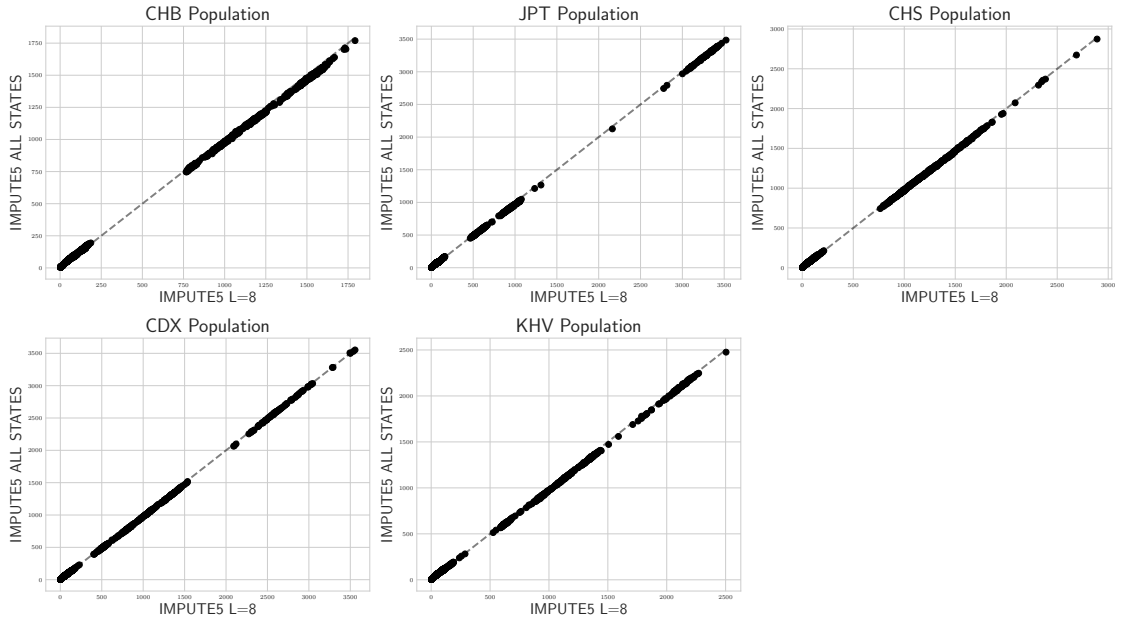

Figure B.4: Concordance between the expected amount of sequence copied from the reference panel (D_i) of IMPUTE5 neighbour select and IMPUTE5 all states for EAS super-population.

Figure B.5: Average expected amount of sequence copied from each of the 1000 Genomes populations for AFR super-population.

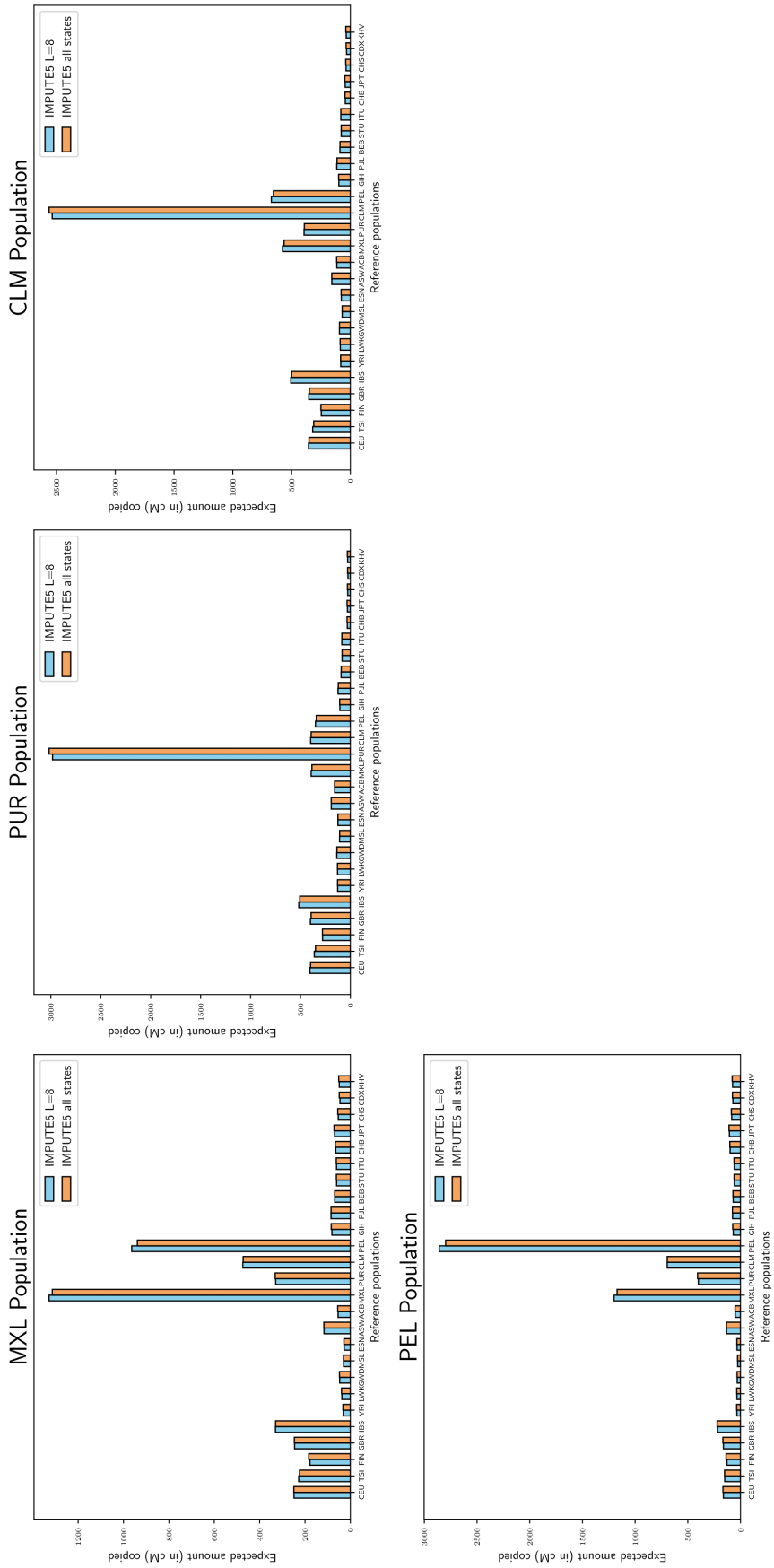

Figure B.6: Average expected amount of sequence copied from each of the 1000 Genomes populations for AMR super-population.

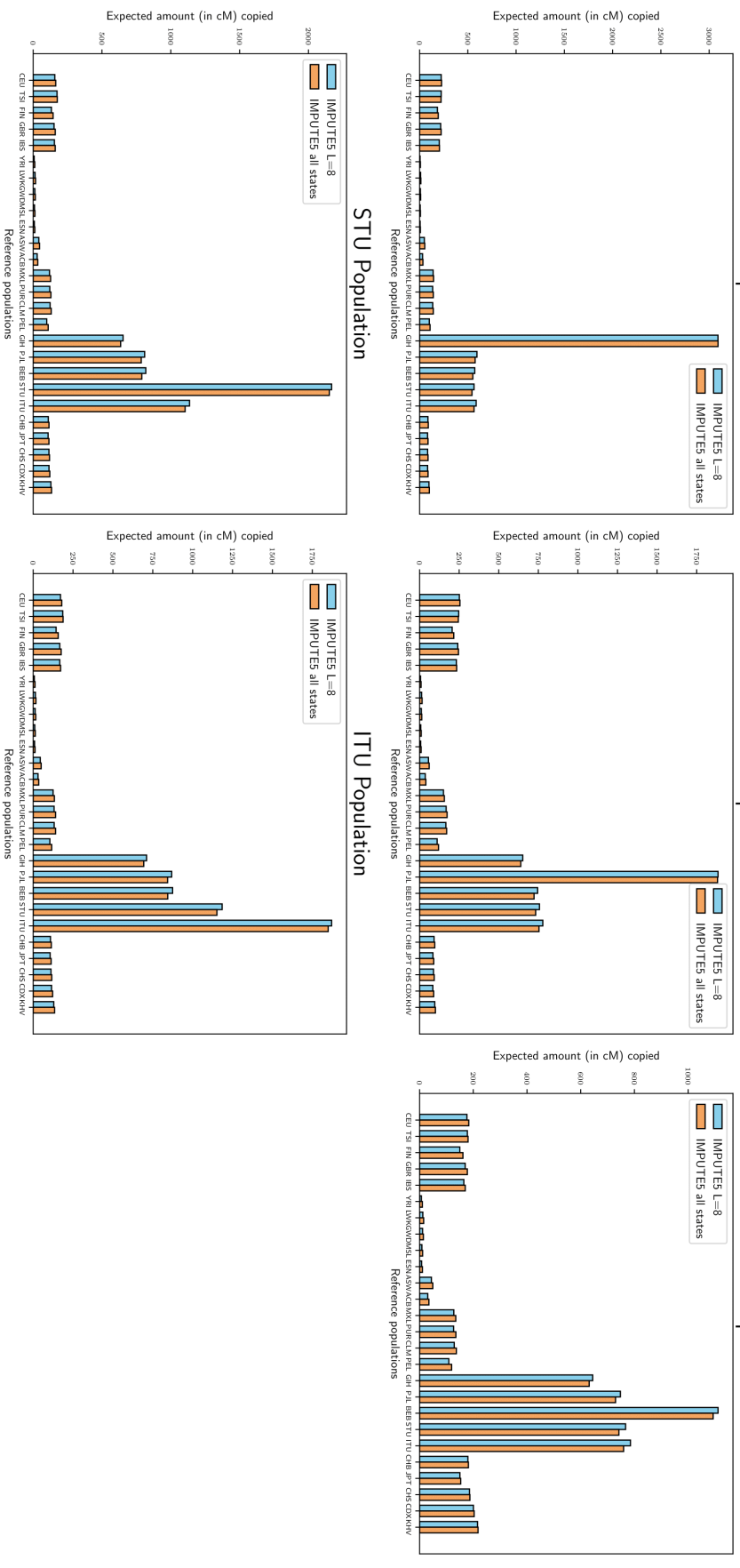

Figure B.7: Average expected amount of sequence copied from each of the 1000 Genomes populations for SAS super-population.

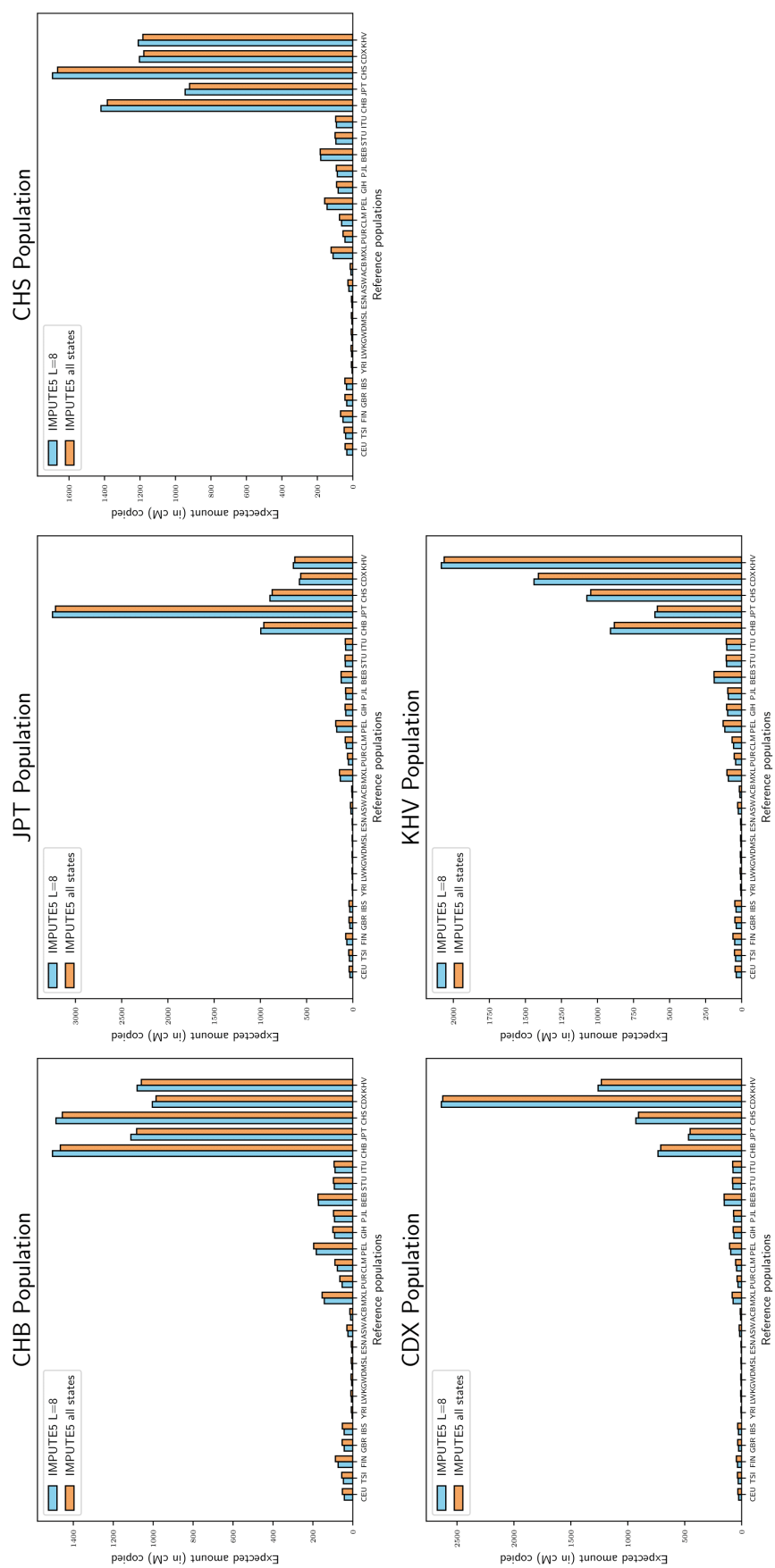

Figure B.8: Average expected amount of sequence copied from each of the 1000 Genomes populations for EAS super-population.

References

Balding, David J. (2006). 'A tutorial on statistical methods for population association studies'. In: *Nature Reviews Genetics* 7.10, pp. 781–791.

Band, Gavin and Jonathan Marchini (2018). 'BGEN: a binary file format for imputed genotype and haplotype data'. In: *BioRxiv*, p. 308296.

Brody, Jennifer A. et al. (2017). 'Analysis commons, a team approach to discovery in a big-data environment for genetic epidemiology'. In: *Nature Genetics* 49, pp. 1560–1563.

Browning, Brian L. and Sharon R. Browning (2016). 'Genotype Imputation with Millions of Reference Samples'. In: *American Journal of Human Genetics* 98.1, pp. 116–126.

Browning, Brian L., Ying Zhou and Sharon R. Browning (2018). 'A One-Penny Imputed Genome from Next-Generation Reference Panels'. In: *American Journal of Human Genetics* 103.3, pp. 338–348.

Burrows, Michael and David J. Wheeler (1994). *A block-sorting lossless data compression algorithm*.

Bush, William S. and Jason H. Moore (2012). 'Chapter 11: Genome-Wide Association Studies'. In: *PLoS Computational Biology* 8.12.

Bycroft, Clare et al. (2018). 'The UK Biobank resource with deep phenotyping and genomic data'. In: *Nature* 562.7726, pp. 203–209.

Cantor, Rita M., Kenneth Lange and Janet S. Sinsheimer (2010). 'Prioritizing GWAS Results: A Review of Statistical Methods and Recommendations for Their Application'. In: *American Journal of Human Genetics* 86.1, pp. 6–22.

Caulfield, Mark et al. (2019). *The National Genomics Research and Healthcare Knowledgebase*.

Chang, Christopher C. et al. (2015). 'Second-generation PLINK: rising to the challenge of larger and richer datasets'. In: *GigaScience* 4.1.

Coleman, Jonathan R. I. et al. (2016). 'Quality control, imputation and analysis of genome-wide genotyping data from the Illumina HumanCoreExome microarray'. In: *Briefings in Functional Genomics* 15.4, pp. 298–304.

Coon, Keith D. et al. (2007). 'A High-Density Whole-Genome Association Study Reveals That *APOE* Is the Major Susceptibility Gene for Sporadic Late-Onset Alzheimer's Disease'. In: *The Journal of Clinical Psychiatry* 68.4, pp. 613–618.

Das, Sayantan (2017). 'Next Generation of Genotype Imputation Methods'. PhD thesis. University of Michigan.

Das, Sayantan et al. (2016). 'Next-generation genotype imputation service and methods'. In: *Nature Genetics* 48.10, pp. 1284–1287.

Delaneau, Olivier et al. (2018). 'Integrative haplotype estimation with sub-linear complexity'. In: *bioRxiv*, p. 493403.

Durbin, Richard (2014). 'Efficient haplotype matching and storage using the positional Burrows-Wheeler transform (PBWT)'. In: *Bioinformatics (Oxford, England)* 30.9, pp. 1266–1272.

Ellingson, Sally R. and David W. Fardo (2016). 'Automated quality control for genome wide association studies'. In: *F1000Research* 5.

Ferragina, Paolo and Giovanni Manzini (2000). 'Opportunistic data structures with applications'. In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 390–398.

Gao, Xiaoyi (2011). 'Multiple testing corrections for imputed SNPs'. In: *Genetic epidemiology* 35.3, pp. 154–158.

Gusfield, Dan (2004). 'An Overview of Combinatorial Methods for Haplotype Inference'. In: *Computational Methods for SNPs and Haplotype Inference*. Ed. by Sorin Istrail, Michael Waterman and Andrew Clark. Red. by Gerhard Goos, Juris Hartmanis and Jan van Leeuwen. Vol. 2983. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 9–25.

Hellenthal, Garrett et al. (2014). 'A Genetic Atlas of Human Admixture History'. In: *Science* 343.6172, pp. 747–751.

Hill, William G., Michael E. Goddard and Peter M. Visscher (2008). 'Data and Theory Point to Mainly Additive Genetic Variance for Complex Traits'. In: *PLOS Genetics* 4.2, e1000008.

Homburger, Julian R et al. (2019). 'Low coverage whole genome sequencing enables accurate assessment of common variants and calculation of genome-wide polygenic scores'. In: *Genome medicine* 11.1, pp. 1–12.

Howie, Bryan N., Peter Donnelly and Jonathan Marchini (2009). 'A Flexible and Accurate Genotype Imputation Method for the Next Generation of Genome-Wide Association Studies'. In: *PLOS Genetics* 5.6, e1000529.

Howie, Bryan et al. (2012). 'Fast and accurate genotype imputation in genome-wide association studies through pre-phasing'. In: *Nature Genetics* 44.8, pp. 955–959.

Huang, Jie et al. (2015). 'Improved imputation of low-frequency and rare variants using the UK10K haplotype reference panel'. In: *Nature Communications* 6.

Kelleher, Jerome, Alison M. Etheridge and Gilean McVean (2016). 'Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes'. In: *PLoS computational biology* 12.5, e1004842.

Kiryluk, Krzysztof (2016). 'Challenges in Rare Variant Association Studies for Complex Kidney Traits: CFHR5 and IgA Nephropathy'. In: *Journal of the American Society of Nephrology* 27.9, pp. 2547–2551.

LaFramboise, Thomas (2009). 'Single nucleotide polymorphism arrays: a decade of biological, computational and technological advances'. In: *Nucleic Acids Research* 37.13, pp. 4181–4193.

Lambert, Jean-Charles et al. (2013). 'Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer's disease'. In: *Nature Genetics* 45.12, pp. 1452–1458.

Lawson, Daniel John et al. (2012). 'Inference of Population Structure using Dense Haplotype Data'. In: *PLOS Genetics* 8.1, e1002453.

Li, Heng (2016). 'BGT: efficient and flexible genotype query across many samples'. In: *Bioinformatics (Oxford, England)* 32.4, pp. 590–592.

Li, Miao et al. (2012). 'Evaluating the effective numbers of independent tests and significant p-value thresholds in commercial genotyping arrays and public imputation reference datasets'. In: *Human genetics* 131.5, pp. 747–756.

Li, Na and Matthew Stephens (2003). 'Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data'. In: *Genetics* 165.4, pp. 2213–2233.

Li, Yun et al. (2010). 'MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes'. In: *Genetic Epidemiology* 34.8, pp. 816–834.

Locke, Adam E. et al. (2015). 'Genetic studies of body mass index yield new insights for obesity biology'. In: *Nature* 518.7538, pp. 197–206.

Loh, Po-Ru et al. (2016). 'Reference-based phasing using the Haplotype Reference Consortium panel'. In: *Nature Genetics* 48.11, pp. 1443–1448.

Marchini, Jonathan (2011). 'Chapter 10 - Genotype Imputation'. In: *Analysis of Complex Disease Association Studies*. Ed. by Eleftheria Zeggini and Andrew Morris. San Diego: Academic Press, pp. 157–175.

— (2019). 'Haplotype Estimation and Genotype Imputation'. In: *Handbook of Statistical Genomics*. John Wiley & Sons, Ltd, pp. 87–114.

Marchini, Jonathan and Bryan Howie (2010). 'Genotype imputation for genome-wide association studies'. In: *Nature Reviews Genetics* 11.7, pp. 499–511.

Marchini, Jonathan, Bryan Howie et al. (2007). 'A new multipoint method for genome-wide association studies by imputation of genotypes'. In: *Nature Genetics* 39.7, pp. 906–913.

Marees, Andries T. et al. (2018). 'A tutorial on conducting genome-wide association studies: Quality control and statistical analysis'. In: *International Journal of Methods in Psychiatric Research* 27.2.

McCarthy, Shane et al. (2016). 'A reference panel of 64,976 haplotypes for genotype imputation'. In: *Nature genetics* 48.10, pp. 1279–1283.

Mills, Melinda C. and Charles Rahal (2019). 'A scientometric review of genome-wide association studies'. In: *Communications Biology* 2.1, pp. 1–11.

Okada, Yukinori et al. (2014). 'Genetics of rheumatoid arthritis contributes to biology and drug discovery'. In: *Nature* 506.7488, pp. 376–381.

Pärn, Kalle et al. (2019). *Genotype imputation workflow v3.0*. URL: `https://www.protocols.io/view/genotype-imputation-workflow-v3-0-xbgfijw` (visited on 16/09/2019).

Pharoah, Paul D. P. et al. (2013). 'GWAS meta-analysis and replication identifies three new susceptibility loci for ovarian cancer'. In: *Nature Genetics* 45.4, 362–370, 370e1–2.

Price, Alkes L. et al. (2009). 'Sensitive Detection of Chromosomal Segments of Distinct Ancestry in Admixed Populations'. In: *PLOS Genetics* 5.6, e1000519.

Pritchard, Jonathan K. and Molly Przeworski (2001). 'Linkage Disequilibrium in Humans: Models and Data'. In: *American Journal of Human Genetics* 69.1, pp. 1–14.

Pulit, Sara L, Sera AJ de With and Paul IW de Bakker (2017). 'Resetting the bar: Statistical significance in whole-genome sequencing-based association studies of global populations'. In: *Genetic epidemiology* 41.2, pp. 145–151.

Rabiner, Lawrence R. (1989). 'A tutorial on hidden Markov models and selected applications in speech recognition'. In: *Proceedings of the IEEE* 77.2, pp. 257–286.

Schönherr, Sebastian et al. (2012). 'Cloudgene: A graphical execution platform for MapReduce programs on private and public clouds'. In: *BMC Bioinformatics* 13.1, p. 200.

Taliun, Daniel et al. (2019). 'Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program'. In: *bioRxiv*, p. 563866.

Tam, Vivian et al. (2019). 'Benefits and limitations of genome-wide association studies'. In: *Nature Reviews Genetics* 20.8, pp. 467–484.

Tewhey, Ryan et al. (2011). 'The importance of phase information for human genomics'. In: *Nature Reviews Genetics* 12.3, pp. 215–223.

The 1000 Genomes Project Consortium (2015). 'A global reference for human genetic variation'. In: *Nature* 526.7571, pp. 68–74.

The International HapMap Consortium (2007). 'A second generation human haplotype map of over 3.1 million SNPs'. In: *Nature* 449.7164, pp. 851–861.

Tishkoff, Sarah A. et al. (1996). 'Global Patterns of Linkage Disequilibrium at the CD4 Locus and Modern Human Origins'. In: *Science* 271.5254, pp. 1380–1387.

Turnbull, Clare et al. (2018). 'The 100 000 Genomes Project: bringing whole genome sequencing to the NHS'. In: *BMJ* 361, k1687.

Turner, Stephen et al. (2011). 'Quality control procedures for genome-wide association studies'. In: *Current Protocols in Human Genetics* Chapter 1, Unit1.19.

Visscher, Peter M. et al. (2017). '10 Years of GWAS Discovery: Biology, Function, and Translation'. In: *The American Journal of Human Genetics* 101.1, pp. 5–22.

Whalen, Andrew et al. (2018). 'Assessment of the performance of hidden Markov models for imputation in animal breeding'. In: *Genetics Selection Evolution* 50.1, p. 44.

Zhang, Jizhun et al. (2015). 'Use of Genome-Wide Association Studies for Cancer Research and Drug Repositioning'. In: *PLOS ONE* 10.3, e0116477.