Methods for phasing and imputation of very low coverage sequencing data

University of Oxford

Warren Winfried Kretzschmar
Linacre College

Supervisor: Prof. Jonathan Marchini

A dissertation submitted in partial fulfilment of the requirements for the degree of
DPhil in Genomic Medicine and Statistics
July 30, 2016

Wo kämen wir hin
Wenn alle sagten
Wo kämen wir hin
Und niemand ginge
Um einmal zu schauen
Wohin man käme
Wenn man ginge

– Kurt Marti, Rosa Loui

Abstract

The introduction of massively parallel short-read sequencing has facilitated rapidly dropping costs of DNA sequencing. This has led to substantial growth in the size of human sequencing projects, with consortia of low coverage sequencing data containing tens of thousands of samples. However, current statistical methods for genotype calling from this data scale poorly with sample size, and are infeasible to use on the largest of current projects. This thesis explores the problem of genotype calling and phasing of large sample sizes of low-coverage sequencing data.

Current methods are applied to call and phase genotypes of the CONVERGE consortium, a data set consisting of very low coverage next-generation sequencing data collected from around 12,000 Chinese women. A genotyping accuracy of 92% as measured by squared Pearson correlation (R^2) against a SNP genotyping chip is achieved for minor allele frequencies >5%, demonstrating that very low coverage sequencing can be used instead of SNP genotyping chips to genotype a study of this size.

A new statistical model is described that allows genotype calling and phasing of low coverage sequencing data in $N(\log N)$ time complexity, where N is sample size, which greatly improves run time compared to current methods. Other adaptations of the model, including a GPU implementation, are also presented.

The new statistical model is used to call and phase genotypes from the largest collection of low coverage sequencing data in the world (about 32,000 Europeans), the Haplotype Reference Consortium (HRC). At a non-reference allele frequency of 0.1% the HRC haplotypes provide a downstream imputation accuracy of up to 64% R^2, compared to an R^2 of 36% when using 1000 Genomes Phase 3 haplotypes, the largest publicly available collection of haplotypes derived from low coverage sequencing.

Finally, a web server has been written to allow small numbers of high coverage whole genome sequenced samples to be phased using the HRC panel. The HRC panel is only available to HRC consortium members, but this web server allows the academic community to gain access to the HRC panel for phasing their own samples.

Acknowledgments

I thank my supervisor Prof Jonathan Marchini for his guidance, support, and for believing in me when things got rough.

I thank Prof Jonathan Flint and Prof Richard Mott for running a nurturing but demanding DPhil program. Without you I don't know where I would be now.

Thank you Kiran Garimella, Robbie Davies, Na Cai, Vicky Hore, Andy Dahl, Kevin Sharp, Olivier Delaneau, Lloyd Elliot, Ryan Christ, Hilary Martin, Holly Trochet, Alex Young, Clare Bycroft, Jerome Kelleher, Liz Batty, Ran Li, and Colin Freeman for many conversations, deep and shallow, and for helping me broaden my knowledge of statistics, programming, and biology. You were the reason I came to work every day.

I thank the members of the CONVERGE consortium and Haplotype Reference Consortium for giving me access to their unique data sets, Richard Durbin for insightful discussions about my work, and Na Cai, Yihan Li, Jingchu Hu, Li Song, Sayantan Das, and Shane McCarthy for being great team workers and a joy to work with.

My thanks go to the Wellcome Trust for their generous funding of my research.

Thanks to Andy Dahl and Ryan Christ for many discussions that helped me refine the mathematical description of the SNPTools algorithm in chapter 3.

I thank Prof Simon Myers and Prof Heather Cordell for examining my thesis and for the helpful discussions we have had over the course of my studies.

The first shall be last. Thank you, Pavitra, for giving me hope when things felt hopeless, and for brightening my day when they did not.

Contents

Chapter 1

Introduction and background

The cost of DNA sequencing has been dropping exponentially (Wetterstrand, 2013) since the sequencing of the human genome in 2001 (Lander, Linton, et al., 2001; Venter et al., 2001). Wide-spread adoption of massively parallel short-read (30-150 base pairs) sequencing, also known as "next-generation sequencing", by sequencing centers in 2008 was the cause of a further substantial drop in sequencing cost (Wetterstrand, 2013). Since then, next-generation sequencing has been replacing older single nucleotide polymorphism (SNP) genotyping microarrays, also known as "SNP chips", as the primary method of discovering genetic variation in human populations. For example, the SNP chip-based International HapMap Project (The International HapMap Consortium, 2003, 2005, 2007, 2010) was succeeded by the next-generation sequencing-based 1000 Genomes Project (The 1000 Genomes Project Consortium, 2010, 2012, 2015), and the UK10K (UK10K Consortium, 2015) project sequenced a third of its samples using next-generation sequencing. In this thesis we focus on a type of next-generation sequencing used in projects such as the 1000 Genomes and UK10K projects called "low coverage" sequencing and explore new methods for using this data to impute genotypes and call haplotypes.

Sequencing coverage is the ratio of the number of bases sequenced to the size

of the genome sequenced, and it is a function of the amount of sequencing reagent supplied to a sequencer during a sequencing run. Although there are no hard rules for categorizing sequencing coverage, a coverage greater than about 40x is generally called "high coverage" and is used when sequence variation needs to be known unambiguously in the parts of the genome accessible by next-generation sequencing. Examples of its use are in medical genomics (Dewey et al., 2014; Parsons et al., 2014) or for validation of other genotyping methods (McCarthy et al., 2015; The 1000 Genomes Project Consortium, 2010, 2015).

Sequencing coverage as high as 7.6x has been considered low coverage sequencing (UK10K Consortium, 2015). As part of the 1000 Genomes Project (1000G), Y. Li, Sidore, et al. (2011) determined that 4x coverage can be used both to efficiently discover new genetic variation in population samples and to make this variation available to other studies for imputation through a haplotype reference panel. Statistical methods for genotype imputation can greatly improve the quality of genotypes in low coverage sequencing (The 1000 Genomes Project Consortium, 2010, 2012), which is why genotype imputation is an important step in any low coverage sequencing work flow (DePristo et al., 2011).

The act of deconvolving haplotypes from genotypes is called phasing or haplotype inference. Although haplotypes are useful by themselves, for example for population genetics and parent of origin effects, they also play an important role as reference panels for genotype imputation in large genetic association studies called genome-wide association studies (GWAS). In the past decade, GWAS have become a popular method of discovering genetic associations with common complex diseases (Welter et al., 2014). Haplotype reference panels derived from low coverage sequencing have been found to increase the power of GWAS (Marchini, B. Howie, et al., 2007; The Wellcome Trust Case Control Consortium, 2007), and for that reason research into improving the size and quality of haplotype reference panels for this purpose has

continued (McCarthy et al., 2015; UK10K Consortium, 2015).

In chapter 2 we explore how the work flows for processing low coverage sequencing data pioneered in the 1000G (DePristo et al., 2011; Y. Li, Sidore, et al., 2011) and by others (Pasaniuc et al., 2012) can be applied to a data set consisting of about 12,000 Chinese women sequenced to 1.7x coverage, roughly ten times as many samples as were part of the first phase of the 1000G (The 1000 Genomes Project Consortium, 2012). We discover that although most of the data analysis process still applies, there are important differences in genotyping accuracy, due to the lower sequencing coverage of this data set compared to the 1000G, and computational cost due to the size of this data set.

The size of low coverage sequencing data sets has grown dramatically since the start of the 1000G. In chapter 3 we describe a novel method for genotype imputation and haplotype calling that seeks to address the problem of poor computational scaling with sample size found in other methods currently used for genotype imputation in low coverage sequencing projects.

In chapter 4 we use this novel method to create a larger version of the 1000G haplotype reference panel in order to improve the quality of genotype imputation into GWAS. This panel was created as part of the Haplotype Reference Consortium (HRC), a data set consisting of low coverage sequencing data of over 32,000 predominantly European samples. We find that my method substantially reduces the computational cost of creating a unified haplotype reference panel from the HRC data, and that for SNPs with low allele frequency the panel improves the genotype imputation accuracy into GWAS compared to the 1000 Genomes and UK10K projects.

A wrinkle in the HRC project is that in contrast to the 1000G and UK10K, some of the participants in the HRC do not allow the sharing of their genetic data with anyone outside the consortium. Chapter 5 describes a service that was created to allow phasing of samples using the HRC as a reference panel without releasing the

panel itself to the public. This service can be accessed through a web site hosted at the Oxford Statistics Department.

1.1 Computational methods for genotype phasing

To phase a series of genotypes is to assign the alleles of every heterozygote genotype of the series of genotypes to a chromosome. The result of phasing is a diplotype, which consists of m haplotypes, where m is the ploidy of the sample. Computational phasing algorithms rely on pooling data from a set of unrelated individuals to infer haplotypes based on patterns in the genotypes created by a shared genetic history between samples.

The first algorithm to derive haplotypes from genotypes at more than two markers (S. R. Browning and B. L. Browning, 2011) was published by Clark (1990), who devised an iterative approach for phasing genotypes by looking for a parsimonious set of haplotypes to explain observed genotypes. The algorithm infers haplotypes by looking for samples that unambiguously imply diplotypes based on other calculated haplotypes. The problem with this approach is that it has no way of assigning a diplotype to a sample for which the diplotype cannot be identified uniquely.

The next evolution in haplotype estimation was the use of an EM algorithm to estimate the most likely haplotype frequencies in a population (Excoffier and Slatkin, 1995). This approach assigns equal prior probability to all possible configurations of haplotypes. Unfortunately, this approach has exponential complexity with the number of markers being phased, and is therefore impractical for phasing multiple genes or even a whole genome.

Another class of models is based on the approximate coalescent (Fearnhead and Donnelly, 2001; N. Li and Stephens, 2003; Stephens and Donnelly, 2000, but see also McVean and Cardin, 2005). One such model that has been used very successfully is

Li and Stephens' model of the conditional distribution of a single haplotype given a set of N other haplotypes. This model estimates the likelihood of a new haplotype conditional on all other haplotypes such that a new haplotype is more likely if it is more closely related to previously observed haplotypes and can be explained as a mosaic of imperfect copies of other haplotypes. This model is an HMM that can be efficiently implemented. Most modern computational phasing methods such as BEAGLE (S. R. Browning and B. L. Browning, 2007), IMPUTE2 (B. N. Howie, Donnelly, and Marchini, 2009), MaCH (Y. Li, Willer, et al., 2010), SHAPEIT (Delaneau, Marchini, and Zagury, 2011; Delaneau, Zagury, and Marchini, 2013), HAPI-UR(Williams et al., 2012), and SNPTools (Wang et al., 2013) use this approach.

PHASEv2 (Stephens, Smith, and Donnelly, 2001; Stephens and Donnelly, 2003; Stephens and Scheet, 2005) was considered the gold standard for phasing (S. R. Browning and B. L. Browning, 2011; Marchini, Cutler, et al., 2006). Unfortunately, PHASEv2 scales poorly with the number of sites phased, which led to the development of other related methods based on the Li and Stephens model.

FastPHASE and BEAGLE phase haplotypes using localized haplotype cluster models, although the way in which they cluster haplotypes is quite different. FastPHASE clusters haplotypes into a pre-specified number of clusters that represent common unobserved haplotypes from which observed haplotypes could have arisen. For sample sizes below 100, fastPHASE is almost as accurate as PHASEv2 and very fast if a small number of clusters is chosen (S. R. Browning and B. L. Browning, 2011). However, fastPHASE does not gain in accuracy with sample size as other models do, because its clusters cannot represent all the diversity encountered in hundreds of samples or more. Increasing the number of clusters leads to a quadratic increase in run time with a much larger scaling factor than other methods such as BEAGLE.

BEAGLE is both faster and more accurate than fastPHASE sample sizes above 1000 samples (S. R. Browning and B. L. Browning, 2007). It uses a graph to locally

cluster haplotypes into a variable number of clusters (hidden states of the HMM) at each site based on linkage disequilibrium (LD) between neighboring sites. In regions of high LD and when there is evidence of many haplotypes, BEAGLE expands the number of haplotype clusters. Across recombination hotspots or when there is evidence of only a few haplotypes, BEAGLE reduces the number of clusters. The purpose of this expansion and reduction of clusters is to create a parsimonious representation of haplotype diversity in the region being phased and to save on run time. The scaling of BEAGLE's compute requirements with sample size depends on haplotype diversity encountered in the region being phased. In practice BEAGLE appears to scale quadratically with sample size up to the low tens of thousands of samples (S. R. Browning and B. L. Browning, 2007; Williams et al., 2012; but see also subsection 3.2.1), although Loh, Palamara, and Price (2015) present data that BEAGLE may scale super-quadratically with larger sample sizes. Nevertheless, for most data sets BEAGLE is fast, accurate, and its implementation is open source and available with an MIT license.

Whereas BEAGLE uses a variable number of hidden states at each site to represent haplotypes, IMPUTE2 and MaCH use a fixed number of hidden states to represent diplotypes at each site in their HMM. IMPUTE2 and MaCH start by initializing all diplotypes with random phase, and use an MCMC algorithm to update each diplotype in turn conditional on all other haplotypes. MaCH estimates per-site recombination rates using an EM algorithm for HMMs called the Baum-Welch algorithm to estimate the recombination rates. IMPUTE2, in contrast, uses a recombination rate map to fix recombination rates. The MaCH and IMPUTE2 HMMs scale quadratically with the number of haplotypes conditioned on when sampling a new diplotype.

To manage computational complexity, both MaCH and IMPUTE2 restrict the set of conditioning haplotypes. MaCH chooses a random subset of haplotypes, whereas at each iteration IMPUTE2 performs a k-nearest neighbor (k-NN) search for haplotypes

that are similar to the haplotype being updated. The distance metric used in the search is the Hamming distance[1]. The idea behind IMPUTE2's selection criterion is that haplotypes that are close on the genealogical tree to the haplotypes being phased are going to be the most informative for the purpose of phasing, and that haplotypes that are close to each other on the genealogical tree are going to be more similar to each other as measured by Hamming distance than other haplotypes. Another way of thinking of the Hamming distance metric is of a way of choosing haplotypes that share long stretches of Identity by State (IBS). The longer such IBS segments, the greater the probability that these segments are shared by decent (IBS) because of a common genealogical history. The idea of using IBS segments for phasing is not new. For example, Kong et al. (2008) describe a rule-based method that relies entirely on IBS segments for phasing. MaCH and Impute2 deliver higher accuracy than BEAGLE, but at a substantially greater computational cost (Delaneau, Marchini, and Zagury, 2011).

SNPTools can best be understood in how it differs from the IMPUTE2 algorithm. Like IMPUTE2, SNPTools also updates a diplotype conditional on a fixed number of current-guess haplotypes. However, in contrast to IMPUTE2, the number of haplotypes conditioned on is four. In a choice reminiscent of the surrogate parent idea by Kong et al. (2008), the SNPTools treats the four conditioning haplotypes as surrogate parents of the diplotype being phased. Before phasing the diplotype of interest, conditioning haplotypes are sampled from all current haplotypes using an MH sampler, whereas IMPUTE2 uses a Hamming distance-based k-NN search to choose conditioning haplotypes. SNPTools also breaks the MCMC scheme used to update diplotypes by sampling diplotypes of all individuals concurrently (instead of sequentially) based on the current guess of all haplotypes. SNPTools is slow and has quadratic run time complexity with sample size, although its phasing accuracy appears to be comparable

[1]The Hamming distance is a metric that is the number of differences between two strings of equal length.

to BEAGLE (Wang et al., 2013). A much deeper description of the SNPTools phasing algorithm can be found in chapter 3.

The best performing phasing algorithms modify the IMPUTE2/MaCH approach by using an HMM with hidden states that represent non-overlapping haplotype segments spanning multiple markers. This allows compression of the conditioning haplotypes into a small number of clusters that are weighted by the number of haplotypes they represent. Both SHAPEIT and HAPI-UR are examples of this approach, and both outperform BEAGLE in speed and phasing accuracy. SHAPEIT2 (Delaneau, Zagury, and Marchini, 2013) incorporates the k-NN search from IMPUTE2 to decrease the number of conditioning haplotypes for an increase in phasing accuracy and speed (Loh, Palamara, and Price, 2015). For more than ten thousand samples, SHAPEIT2 spends the majority of its time in the k-NN search, the implementation of which has quadratic complexity. O'Connell, Sharp, et al. (2016) present an alternate haplotype clustering approach that reduces SHAPEIT2 complexity to $O(N \log N)$, by replacing the k-NN search with a haplotype partitioning scheme based on k-means clustering.

When pedigrees are (partially) available, the performance of SHAPEIT2 can be improved using an algorithm called duoHMM (O'Connell, Gurdasani, et al., 2014). This method is designed to refine high quality haplotypes produced by another phasing program such as SHAPEIT2 or BEAGLE using pedigree information. DuoHMM can correct haplotype switch errors and identify genotyping errors using an HMM that models the gene flow of parents to their children given the supplied (observed) haplotypes. In pedigrees consisting of at least three siblings or samples from three generations obtained from nominally unrelated and more closely related populations, (O'Connell, Gurdasani, et al., 2014) show that the number of recombinations identified in each chromosome corresponds almost perfectly to the number of recombinations expected based on genetic maps created by the HapMap project.

1.2 Genotype imputation in genome-wide association studies

In its simplest form, a genetic association study is a case-control study where case-control status is correlated to the existence of an allele at a site of interest in the study samples using a 2x2 contingency table (Lander and Schork, 1994). In a genome-wide association study (GWAS), a genetic association is tested at a large selection (typically several hundreds of thousands) of variants across the whole genome.

GWAS became popular in the wake of a recognition that linkage and candidate gene approaches were underpowered for discovering common mutations of moderate effect size (Collins, Guyer, and Chakravarti, 1997; Khoury and Yang, 1998). Khoury and Yang (1998) showed that GWAS, while still technically infeasible in the late nineties, could attain the power necessary to detect these mutations. Subsequently, efforts were made to discover SNPs genome wide (The International HapMap Consortium, 2003; The International SNP Map Working Group, 2001). Parallel investigations into statistical modeling of haplotypes at the human genome scale (for example, Stephens, Smith, and Donnelly, 2001; Stephens and Scheet, 2005; but see also Marchini, Cutler, et al., 2006 for a comprehensive comparison) allowed haplotyping of SNPs (Marchini, Cutler, et al., 2006) for the first release of HapMap (The International HapMap Consortium, 2005).

The need to be able to impute missing genotypes while phasing thousands of samples, coupled with the development of a hidden Markov model-based (HMM) (Rabiner, 1989) approach of modeling haplotype sharing between samples (N. Li and Stephens, 2003), led to multiple models that were able to phase and impute missing genotyping array data on a chromosome level. These models included fastPHASE (Scheet and Stephens, 2006), BEAGLE (S. R. Browning and B. L. Browning, 2007),

IMPUTE (Marchini, B. Howie, et al., 2007), and MaCH (Y. Li, Willer, et al., 2010)[2]. The IMPUTE paper demonstrated that imputing genotypes into SNP genotype data from newly available haplotype reference panels (The International HapMap Consortium, 2005, 2007) increases power to detect association signals in GWAS (Marchini, B. Howie, et al., 2007; The Wellcome Trust Case Control Consortium, 2007). IMPUTE uses a diploid version of the N. Li and Stephens (2003) model to phase each study sample from a reference panel and impute missing genotypes based on inferred haplotype copying states.

Further developments in genotype imputation into SNP chip-based GWAS data sets have built on pre-phased study samples (B. Howie, Fuchsberger, et al., 2012). In this approach, a fast and accurate phasing program such as SHAPEIT2 is used to phase study samples at genotyped sites, and a haploid imputation program is used to impute missing data. IMPUTE, IMPUTE2, and MaCH/Minimac (Fuchsberger, Abecasis, and Hinds, 2014) do this using a haploid version of their diploid HMMs. The motivation behind this technique was to cut down the cost of imputing missing data from multiple reference panels. However, pre-phasing also allows the use of haploid models that are not easily adapted to the diploid phasing problem.

One such example is PBWT (Durbin, 2014), which uses a computational technique called the Burrows-Wheeler transform to reorder haplotypes such that similar haplotypes are stored together. This greatly increases the compressibility of haplotypes, but it can also be used to find haplotypes that are identical by state (IBS) for long stretches. The longer such shared stretches of DNA, the greater the chance that these stretches also share a recent genetic history. If so, then such stretches are Identical By Descent (IBD), and one haplotype can be used to impute missing alleles in the other inside such an IBD region.

[2]MaCH was available several years before its publication in 2010. Papers published before 2010 cite the 2006 American Society for Human Genetics meeting abstract for MaCH: "Li Y, Ding J, Abecasis GR. Mach 1.0: Rapid Haplotype Reconstruction and Missing Genotype Inference. American Journal of Human Genetics. 2006;79:S2290".

1.3 Genotype Likelihood calculation from low coverage sequencing

A read, as produced by a short-read next-generation sequencing machine, consists of a series of base calls, each with an associated base quality score. A base quality score Q encodes the probability e that the base call made for a base in a sequencing read is erroneous. e is estimated from the images used by the sequencer to make base calls. A common encoding of e is the Phred score $Q = -10 \log_{10} e$.

However, reads come with no context as to the location in the genome that they were sequenced from. Read aligners (or mappers) are algorithms that are used to map each read to a reference genome. Using these mapped reads, genotype callers can be used to call the genotype of every sequenced position in the reference genome.

In low coverage sequencing, the number of reads is generally not sufficient to accurately call genotypes directly, and instead a statistical model such as the Bayesian Binomial Mixture Model implemented in SNPTools (Wang et al., 2013) is used to calculate genotype likelihoods (GLs) at sites determined to be likely SNPs by algorithms called variant callers.

A GL is the probability of observing an individual's reads at a position given the underlying (unknown) genotype. GLs across all sites and all samples of a data set are then used by statistical models that jointly model all haplotypes in the data set to estimate genotypes and call phased haplotypes. These are the broad strokes of what members of the 1000G consortium worked out as the best way to call SNP genotypes from low coverage (around 4x coverage) sequencing data (DePristo et al., 2011; The 1000 Genomes Project Consortium, 2012).

The problem of estimating and phasing genotypes from GLs has been approached by modern methods (B. L. Browning and Yu, 2009; Delaneau and Marchini, 2014; H. Li, 2011; Y. Li, Sidore, et al., 2011; McKenna et al., 2010; The 1000 Genomes

Project Consortium, 2010, 2012, 2015; Wang et al., 2013) in a Bayesian framework. Let D_{il} be the sequenced bases of individual i at site l, and let $G_{il} \in \{0, 1, 2\}$ be individual i's unobserved genotype at site l. For a sample of N individuals sequenced at L sites, the problem that these and other methods solve is finding probability of the unobserved genotypes given the observed data

$$P(G|D) = \frac{P(D|G)P(G)}{\sum P(D|G)P(G)} \tag{1.1}$$

where $P(D|G)$ is the probability of all observed reads given all unobserved genotypes, and $P(G)$ is the prior probability of all genotypes.

Under the assumption that sequencing reads generated for an individual are independent of the genotypes of other individuals given the genotypes of that individual, we can simplify the previous equation to

$$\begin{aligned} P(G|D) &\propto P(D_i|D_{-i}, G_i, G_{-i})P(D_{-i}|G_i, G_{-i})P(G) \\ &= P(D_i|G_i)P(D_{-i}|G_{-i})P(G) \\ &= \left(\prod_{i=1}^{N} P(D_i|G_i)\right) P(G), \end{aligned} \tag{1.2}$$

where D_{-i} and G_{-i} are respectively the observed reads and unobserved genotypes of all individuals other than i.

1.4 Site-independent genotype calling from GLs

Site-independent genotype calling algorithms assume that each site's genotype is independent of the genotypes of all other sites. While this assumption is not true because of genetic linkage (or linkage disequilibrium) between sites, description of site independent calling methods serves as a motivation for linkage disequilibrium (LD) aware genotype calling methods.

For an individual i, if we assume that reads at a site l are observed independently of all other reads given individual i's genotype at site l, then

$$P(D_i|G_i) = \prod_{l=1}^{L} P(D_{il}|G_{il}), \tag{1.3}$$

where $P(D_{il}|G_{il})$ is a GL. This leads to the definition of a genotype probability as

$$P(G_{il}|D_{il}) = \frac{P(D_{il}|G_{il})P(G_{il})}{\sum_{g=0}^{2} P(D_{il}|G_{il} = g)P(G_{il} = g)}. \tag{1.4}$$

Genotype calls can be made from genotype probabilities by for example choosing the genotype with the highest genotype probability.

After GLs have been called one sample at a time, the focus of site-independent genotype calling algorithms is to pool sequencing information across multiple samples to calculate $P(G_{il})$, the prior distribution on individual i's genotypes at site l. Samtools, The Genome Analysis Toolkit (GATK), and Bcftools, follow the same general approach, but with some variation.

1.4.1 Samtools

Samtools (H. Li, 2011) follows the strategy of estimating genotype frequencies as the set of frequencies that maximize the probability of d, the data observed at a site l for all samples, where d_i is the i^{th} individual's data. The likelihood to maximize is

$$\mathcal{L}(\xi_0, \ldots, \xi_m) = \prod_{i=1}^{N} \sum_{g=0}^{m} P(d_i|g)\xi_g, \tag{1.5}$$

where ξ_g is the genotype frequency of the genotype containing g non-reference alleles. For diploid organisms such as humans the maximum number of non-reference alleles a genotype can contain, m, is 2. The sum inside the product of this likelihood makes direct maximization difficult, and indicates the use of an EM algorithm for finding

the genotype frequencies. The genotype frequencies are treated as latent variables.

1. Make initial guesses for $\hat{\xi}_g$

2. *Expectation step*: compute the expected values of γ_g

$$\hat{\gamma}_g = \frac{1}{N}\sum_{i=1}^{N}\frac{P(d_i|g)\hat{\xi}_g}{\sum_{g'}P(d_i|g')\hat{\xi}_{g'}}, \quad g, g' = 0, 1, 2. \tag{1.6}$$

3. *Maximization step*: compute the updated genotype frequencies

$$\hat{\xi}_g = \hat{\gamma}_g, \quad g = 0, 1, 2. \tag{1.7}$$

4. iterate steps 2 and 3 until convergence.

Samtools then simply takes $P(G_{il} = g) = \hat{\xi}_g$.

1.4.2 Bcftools

Recently, the genotyping functionality of Samtools has been transferred to another tool called Bcftools, and some changes have been made to the genotyping model (Danecek, Schiffels, and Durbin, 2014). Bcftools uses a multi-allelic-aware genotyping method to call genotypes from GLs. Furthermore, instead of estimating latent genotype frequencies as in Equation 1.5 for $P(G_{il})$, Bcftools calculates $P(G_{il})$ as the product of allele frequencies of the alleles of G_{il}, which assumes Hardy-Weinberg equilibrium (HWE). The added assumption of HWE results in a speedup compared to the Samtools method.

Danecek, Schiffels, and Durbin (2014) define a genotype as g, an ordered pair of two alleles, where $g_1, g_2, x, y \in A = \{\text{A, C, G, T}\}$. For each allele, D_x bases are observed for individual i at a site with base qualities Q_{ik}^x, where $1 \leq k \leq D_x$. Here Q_{ik}^x is defined as the probability that the k^{th} base of individual i is not in error. The

probability of an error is defined in section 1.3. The allele frequencies are calculated as

$$f_x = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_k Q_{ik}^x}{\sum_{k,y} Q_{ik}^y}. \tag{1.8}$$

This estimates the overall allele frequency of x as the average of the per-sample allele frequencies. Danecek, Schiffels, and Durbin (2014) want their approach to work with samples of widely varying sequencing depth. That is why they choose to give every sample equal weight in estimating allele frequency. However, Danecek, Schiffels, and Durbin (2014) do not explore the statistical properties of this estimator.

For a given set of alleles $S \subseteq A$, the relative allele frequency is defined as

$$f_{x|S} = \frac{f_x}{\sum_{y \in S} f_y}. \tag{1.9}$$

Under the assumption of HWE, Danecek, Schiffels, and Durbin (2014) calculate

$$P(G = g) = f_{g_1|S} f_{g_2|S}. \tag{1.10}$$

Although Danecek, Schiffels, and Durbin (2014) also estimate S, we shall assume that S has already been estimated during SNP calling.

1.4.3 The GATK's Unified Genotyper

Although aligned reads usually have base call error probabilities and mapping error probabilities available, the Unified Genotyper assumes that any supplied bases are correct. This assumption requires stringent filtering of bases before they are supplied to the Unified Genotyper. For example, DePristo et al. (2011) pre-process input data to the Unified Genotyper by filtering out bases with a Phred quality score (see section 1.3) less than 20, and reads with a mapping Phred quality score less than 20.

To call genotypes, The Genome Analysis Toolkit (GATK) first estimates the

non-reference allele count Q as $Q = \sum_{i=1}^{N} Q_i$, where $Q_i \in \{0, 1, 2\}$ is the number of non-reference alleles in individual i's genotype. DePristo et al. (2011) estimate the probability that $Q = q$ as:

$$\begin{aligned}
P(Q = q|D) &= \frac{P(Q = q)P(D|Q = q)}{\sum_y P(D|Q = y)} \\
P(Q = q) &= \begin{cases} \theta/q & \text{if } q > 0 \\ 1 - \theta \sum_{i=1}^{2N} 1/i & \text{otherwise} \end{cases} \\
P(D|Q = q) &= \sum_{GT \in \Gamma} \prod_{i}^{N} P(D_i|GT_i) \\
\Gamma &= \{GT \text{ where } (\sum_i Q_i) = q\},
\end{aligned}$$

where θ is the genome wide population specific heterozygosity of 0.8×10^{-3} in European CEPH individuals, Γ is the set of all possible genotype assignments for N individuals such that $Q = q$. DePristo et al. (2011) describe $P(Q = q)$ as "the infinite-sites neutral expectation to observe q alternative alleles in $2N$ chromosomes with heterozygosity of θ", but no literature is cited, and it is unclear what work this assertion is derived from.

Although DePristo et al. (2011) do not specify how they call genotypes once Q has been estimated, one approach is to assume HWE and calculate $P(G)$ in the same way as Danecek, Schiffels, and Durbin (2014). If we define the maximum posterior allele frequency as

$$f_{\max} = \frac{1}{N} \operatorname*{argmax}_q P(Q = q|D), \tag{1.11}$$

then we can calculate the prior probability

$$P(G = g) = \begin{cases} f_{\max}^2 & \text{if } g = 0 \\ 2f_{\max}(1 - f_{\max}) & \text{if } g = 1 \\ (1 - f_{\max})^2 & \text{if } g = 2 \end{cases} \tag{1.12}$$

1.5 LD-based genotype calling and phasing from genotype likelihoods

LD-based genotype calling methods are usually extensions of genotype phasing algorithms that have been extended to handle GLs. THUNDER (Y. Li, Sidore, et al., 2011) and SNPTools (Wang et al., 2013) add an extra emission layer to a MaCH/IMPUTE2 style HMM, BEAGLE (B. L. Browning and S. R. Browning, 2009; B. L. Browning and Yu, 2009) adds a layer to its diploid HMM emissions, and Delaneau and Marchini (2014) add an emission layer to SHAPEIT2 based on GLs. An exception to this rule is MVNCall (Menelaou and Marchini, 2013), which uses a multivariate normal model to call genotypes from GLs when a phased haplotype scaffold is available. The SNPTools algorithm is described in detail in chapter 3. The other methods are described in further detail below.

1.5.1 BEAGLE

BEAGLE is a popular genotype and haplotype caller for large low-coverage sequencing studies (CONVERGE Consortium, 2015; Pasaniuc et al., 2012; The 1000 Genomes Project Consortium, 2012, 2015; UK10K Consortium, 2015). The genotype phasing algorithm (S. R. Browning and B. L. Browning, 2007) relies on an iterative model building and haplotype sampling scheme that is initialized with randomly phased genotypes. Initial haplotypes are used to fit a haplotype HMM, and a diplotype HMM implied by the haplotype HMM under the assumption of HWE is sampled from to generate a new set of haplotypes consistent with observed genotypes. Once new haplotypes have been sampled, a new iteration of model fitting and haplotype sampling is started. In the last iteration the most likely haplotypes are derived from the diplotype HMM using the Viterbi algorithm instead of backward sampling. The haplotype HMM clusters haplotypes based on local LD, which greatly reduces the

number of states of the HMM.

The GL-based BEAGLE phasing algorithm (B. L. Browning and S. R. Browning, 2009; B. L. Browning and Yu, 2009) works in the same fundamental way. However, when haplotypes are sampled from the diploid HMM, GLs instead of genotypes are used as emissions for the HMM states. It is unclear from the BEAGLE publications (B. L. Browning and S. R. Browning, 2009; B. L. Browning and Yu, 2009; S. R. Browning and B. L. Browning, 2007) how the GL-based phasing algorithm initializes haplotypes. It should be noted that the original phasing algorithm benefits from restricting HMM states through hard genotype calls. In the case of phasing from GLs, more genotypes are possible, which leads to a larger HMM state space and longer running times.

1.5.2 THUNDER

THUNDER (Y. Li, Sidore, et al., 2011) is an extension of the MaCH model to handle GLs. The extension changes the HMM emissions by treating emitted genotypes as a hidden variable and adding the observed reads as a layer on top of the genotype emissions. A state S of two copying haplotypes (x, y), where $x, y \in \{1, \ldots, 2(N-1)\}$, emits a genotype G that now emits observed reads D:

$$P(D|S = (x, y)) = P(D|G = g)P(G = g|S = (x, y)), \tag{1.13}$$

where $P(D|G = g)$ is a genotype likelihood.

Although THUNDER alone produces poorer haplotypes than BEAGLE (Pasaniuc et al., 2012), it has been observed in the 1000G that THUNDER is slow to converge on low coverage sequencing data, and that initializing THUNDER from haplotypes created by BEAGLE leads to improved haplotypes (Delaneau and Marchini, 2014). This is the configuration that was used for the first phase of 1000G (The 1000 Genomes

Project Consortium, 2012). This idea of creating initial haplotypes with BEAGLE and using them as input for another phasing program continues to be popular in other low coverage sequencing projects (CONVERGE Consortium, 2015; Delaneau and Marchini, 2014; McCarthy et al., 2015; Pasaniuc et al., 2012; UK10K Consortium, 2015). However, with the exception of McCarthy et al. (2015) and Delaneau and Marchini (2014), these projects only keep genotype information supplied by BEAGLE and rely on SHAPEIT2 to rephase the genotypes.

1.5.3 MVNCall

MVNCall (Menelaou and Marchini, 2013) uses a multivariate normal distribution to phase GLs onto a previously phased haplotype scaffold. This method is only applicable when high quality genotypes from a SNP chip or equivalent are available that can be used as a scaffold. Sites that are not part of the scaffold, but for which GLs are available, are then phased independently of each other based on LD between a site and the scaffold. For each site not on the scaffold, an MCMC sampler is used to iteratively update allele pairs of individuals based on covariance between haplotypes at scaffold sites and the site being updated, and GLs. These MCMC updates are very fast and parallelizable. Although the multivariate normal approximation is a simple one, genotype imputation accuracy of African samples presented by Menelaou and Marchini (2013) indicate that when a reliable haplotype scaffold is available, this approach outperforms BEAGLE, THUNDER, and SNPTools.

1.5.4 SHAPEIT2

First BEAGLE is used to estimate genotype probabilities and call haplotypes from the data set GLs. Delaneau and Marchini (2014) then follow the THUNDER approach and initialize with haplotypes derived from BEAGLE. BEAGLE genotypes with a genotype probability above 0.995 are hard called as that genotype, leaving 3% of

genotypes uncalled. During phasing, uncalled genotypes are imputed with the help of GLs. A haplotype scaffold like the one used by MVNCall can also be used to constrain the SHAPEIT2 model, which further improves phasing and to a lesser extent genotyping accuracy (Delaneau and Marchini, 2014). Delaneau and Marchini (2014) further show that their method outperforms BEAGLE and BEAGLE + THUNDER on the 1000G Phase 1 data. This version of SHAPEIT2 was used in the creation of the 1000G Phase 3 haplotype panel (The 1000 Genomes Project Consortium, 2015).

However, the SHAPEIT2 model can only handle data sporadic missing data (Delaneau, Zagury, and Marchini, 2013). In practice this means that sites and samples with more than 5% missing genotypes are filtered out by SHAPEIT2 before running. Delaneau and Marchini (2014) did not explore sequencing data with less than 4x coverage. Hence, for lower coverage sequencing data it is unclear if the threshold of 0.995 on BEAGLE genotype probabilities will hard call enough genotypes to stay under the 5% maximum of missing genotypes allowed by SHAPEIT2.

Chapter 2

The CONVERGE data set

2.1 Introduction

Between 2000 and 2011, Major depressive disorder (MDD) was the leading cause of disability adjusted life years (DALYs[1]) lost in the world (Department of Health Statistics and Information Systems, 2013). It is a complex disease caused both by environmental and genetic effects, with twin studies estimating the heritability as $31\% - 47\%$ (Sullivan, Neale, and Kendler, 2000). Despite having this major genetic component, no genome-wide association study for MDD has been able to find a genome-wide significant result (Boomsma et al., 2008; Kohli et al., 2011; Lewis et al., 2010; Muglia et al., 2010; Rietschel et al., 2010; Shyn et al., 2011; Sullivan, Geus, et al., 2009; Wray et al., 2012) and replicate it in an independent cohort. Similarly, a mega-analysis of 9,420 cases and 9,519 controls also found no result with genome-wide significance (Ripke et al., 2013).

Two reasons why previous genetic association studies may not have found consistent signals for MDD are that it may have heterogeneous environmental and genetic

[1]Disability adjusted life years (DALYs) is a measure of disease burden that takes both early death and morbidity due to a disease into account. DALYs = YLL + YLD, where YLL are years of life lost due to dying early, and YLD are years lived with disability multiplied by a factor (the disease weight) that quantifies how much a disease affects a person's quality of life.

causes, and that MD may well be a complex trait with many genetic components that have small effect sizes (Ripke et al., 2013). CONVERGE (China, Oxford and VCU Experimental Research on Genetic Epidemiology) is a genome-wide genetic association study of MDD in 11,670 Chinese women that seeks to address phenotype heterogeneity by only sampling women with extreme, recurrent cases of MDD, and lack of sensitivity to multiple types of genetic variation by using sparse (<2x coverage) next generation sequencing to genotype samples.

It is quite uncommon for a GWAS with as many samples as CONVERGE to have its samples genotyped using short read whole genome sequencing. For example, all studies in the mega-analysis for MDD (Ripke et al., 2013) genotyped their samples on a SNP genotyping chip. The same is true for other major GWAS consortia for common diseases (Morris et al., 2012; Nikpay et al., 2015). It is unclear from (CONVERGE Consortium, 2015) why 1.7x coverage sequencing was chosen for the CONVERGE project. However, studies published around the time when analysis of the CONVERGE sequencing data began indicate that (i) genotypes called with The Genome Analysis Toolkit (GATK) and BEAGLE in 61 samples sequenced to 1x coverage have low genotype discordance with genotypes from 60x coverage short read sequencing (DePristo et al., 2011), (ii) genotypes with a minor allele frequency (MAF) $> 0.5\%$ and called from simulated reads (32 base pairs) at 2x coverage in 2000 samples have similar genotyping accuracy as genotypes called from simulated reads at 4x coverage, when measured by R^2, and (iii) using a reference panel with BEAGLE when performing genotype imputation of 1x coverage sequencing data in 100 simulated samples can further improve genotyping accuracy (Pasaniuc et al., 2012).

According to work done by the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2010, 2012) the most accurate genotype calls can be obtained by first applying the BEAGLE (B. L. Browning and Yu, 2009; S. R. Browning and B. L. Browning, 2007) method to obtain an initial set of estimated haplotypes. These haplotypes

are then used to initialize a program such as IMPUTE2 (B. N. Howie, Donnelly, and Marchini, 2009), MaCH (Y. Li, Willer, et al., 2010), or SHAPEIT2 (Delaneau, Zagury, and Marchini, 2013) to further refine the estimates. However, using this strategy on the CONVERGE cohort would require a large amount of computational resources, as this cohort is one of the largest whole genome sequenced cohorts in the world.

The aim of this chapter is to explore if the findings of DePristo et al. (2011) and Pasaniuc et al. (2012) can be applied to a study with a substantially larger sample size than has been investigated before. In so doing an analysis pipeline is developed to estimate genotypes and phase haplotypes from aligned 1.7x coverage Illumina next-generation sequencing reads for the CONVERGE consortium using standard methods in the field. A GWAS run on these genotype calls run by the CONVERGE Consortium (2015) discovered the first two replicated genetic hits for MDD.

2.2 Genotype calling

The CONVERGE genotype call set was generated over four rounds (see Figure 2.1 and Figure 2.2). Round 1 was a pilot designed to determine good parameters for imputing the genotypes of most of the CONVERGE samples available at Oxford (9,300 samples) in round 2. I performed rounds 1 and 2, but due to export restrictions on the remainder of the aligned sequencing reads by the Chinese government, it was not possible for me to run my pipeline on the complete data set. Instead, my analysis pipeline was run on the whole cohort (just under 11,670 samples) by researchers at the Beijing Genomics Institute (BGI) in Shenzhen, China, for round 3. A separate imputation run (round 4) was also performed by the BGI at all discovered sites, but without using a reference panel. The BGI then merged that call set with the genotypes from round 3 to create an integrated genotype call set by replacing the

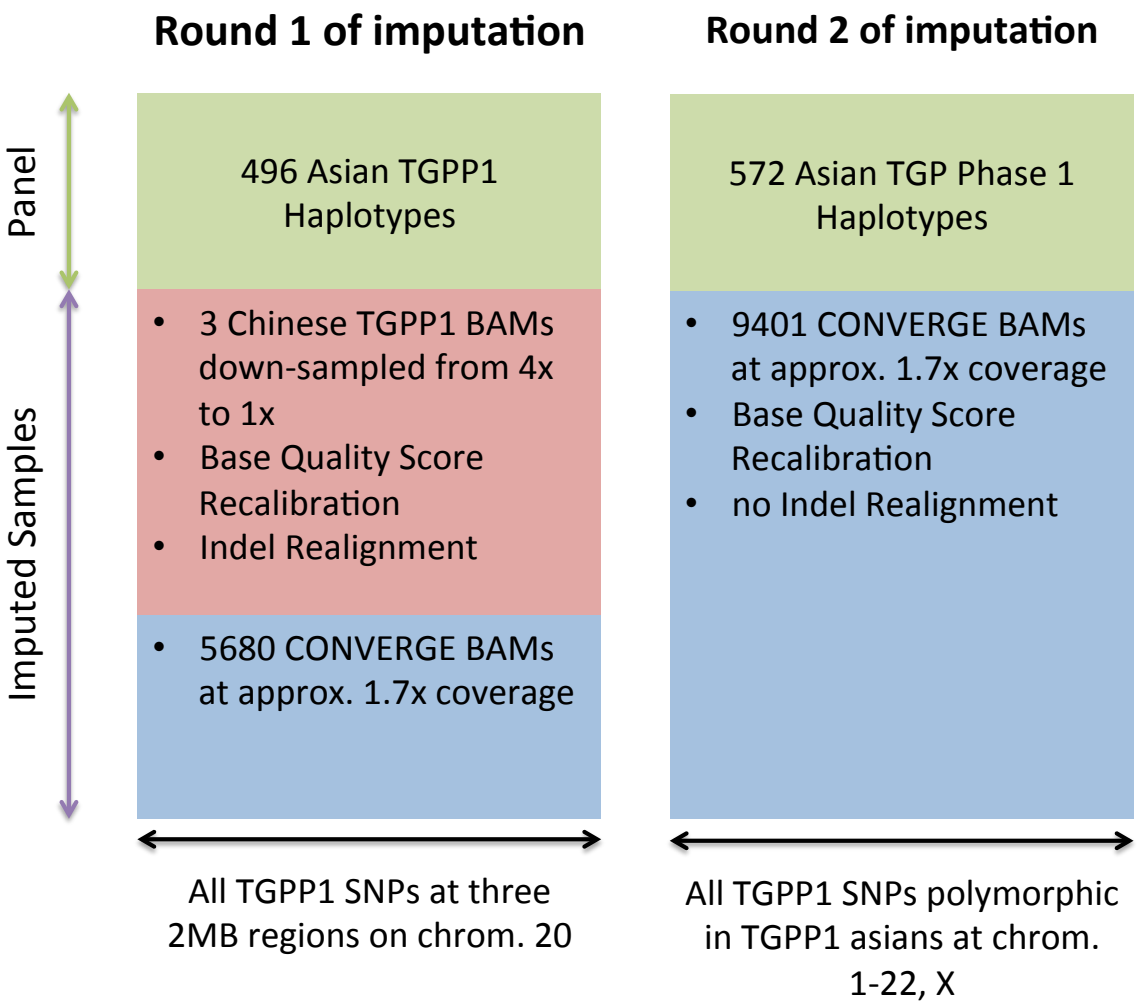

Figure 2.1: *Description of imputation rounds for CONVERGE project: Rounds 1 and 2.* In round 1 the downsampled BAM files of three 1000 Genomes Project Phase 1 (TGPP1) samples were included to asses genotyping accuracy.

calls in round 4 that overlap with round 3 by the calls from round 3. The complete analysis pipeline is described schematically in Figure 2.3.

While working with the CONVERGE data it became clear that a straight application of the methods described by Pasaniuc et al. (2012) was not going to work for the CONVERGE data. For example, just running BEAGLE one chromosome at a time, as was done by Pasaniuc et al. (2012) on a smaller data set, caused BEAGLE to use more memory than was available on the entire server on the round 1 data. Another problem was run time. Although it was clear from informal time keeping

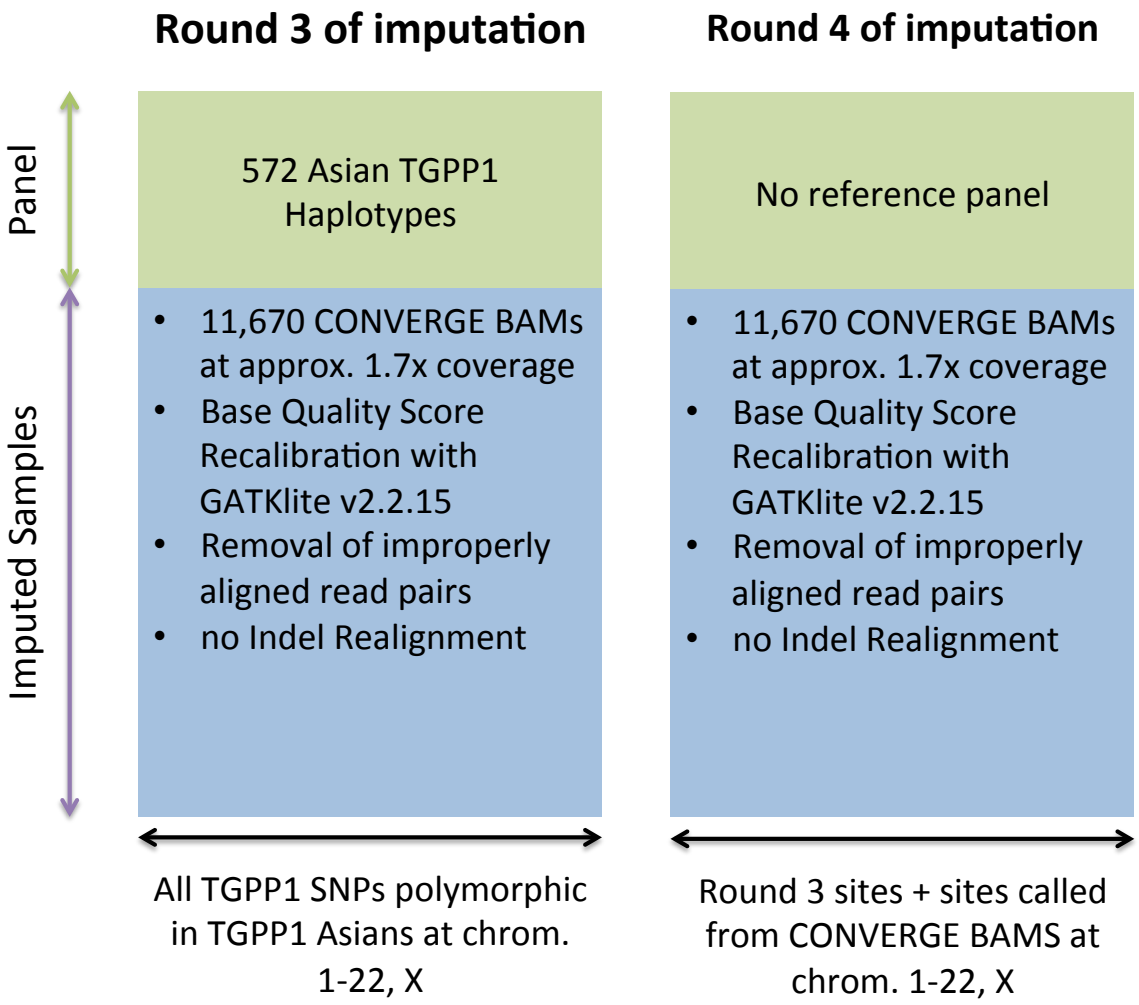

Figure 2.2: *Description of imputation rounds for CONVERGE project: Rounds 3 and 4.*

during round 1 that BEAGLE run time was going to be an issue, on the round 2 samples it was estimated that running BEAGLE on the whole genome using the complete 1000 Genomes Project Phase 1 (1000GP1) haplotypes as a reference panel, as done by Pasaniuc et al. (2012), would take about 33,125 CPU days to complete. As this was more compute time than we had available for the CONVERGE project, ways to reduce BEAGLE run time were investigated. Due to the large run time requirements of BEAGLE, ways of reducing BEAGLE's memory footprint in order to run BEAGLE on the low memory nodes of the cluster were also investigated.

The principal findings from round 1 were that using a reference panel consisting of 496 1000GP1 East Asian (ASN) haplotypes increases imputation accuracy, as had been suggested by Pasaniuc et al. (2012), and that BEAGLE run time and memory usage could be reduced without appreciable loss in genotype imputation accuracy. With round 2 we established that my analysis pipeline provides accurate genotype calls on a nearly complete cohort, and that BEAGLE-based genotype imputation can be performed on the round 2 data using an acceptable amount of compute time. We estimate the amount of compute needed to run BEAGLE was about 3,900 and 1,560 CPU days to respectively impute genotypes at all 1000GP1 SNPs and only the subset of SNPs polymorphic in ASN haplotypes. In rounds 3 and 4 the final integrated genotype call set was created.

2.2.1 Materials and Methods

2.2.1.1 Sample collection

The CONVERGE data set consists of phenotypes and DNA sequences of 11,670 samples, approximately half of which are cases of recurrent MDD, and half controls. Cases were recruited at 58 provincial mental health centers in 45 cities in 23 provinces of China. Controls were recruited from patients undergoing minor surgery at general hospitals (37%) and local community centers (63%). To increase the genetic homo-

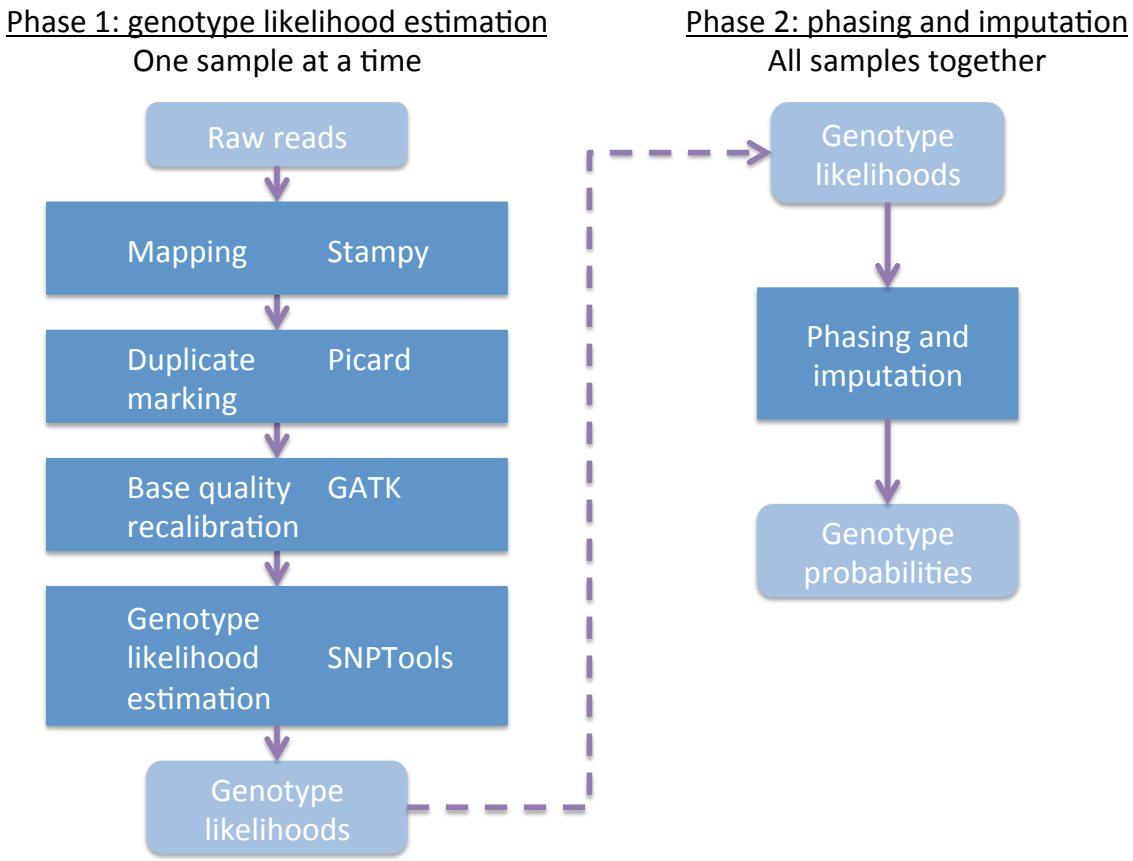

Figure 2.3: *Schematic representation of CONVERGE analysis pipeline.* Mapping and duplicate marking was performed by collaborators at the BGI.

geneity of the study, all cases and controls are female and have four Han Chinese grandparents. The sample size was chosen based on the expectation at the time of the study design (2007) that loci with an odds ratio of 1.2 and above are likely to exist for major depression. Cases were excluded if they had a pre-existing history of bipolar disorder, psychosis or mental retardation. Cases were aged between 30 and 60 and had two or more episodes of MDD meeting DSM-IV criteria (APA, 1994) with the first episode occurring between 14 and 50 years of age, and had not abused drugs or alcohol before their first depressive episode. All subjects were interviewed using a computerized assessment system. Interviewers were postgraduate medical students, junior psychiatrists or senior nurses, trained by the CONVERGE team for a minimum of one week. The diagnosis of MDD was established with the Composite International Diagnostic Interview (CIDI) (WHO lifetime version 2.1; Chinese version), which utilized DSM-IV criteria. The interview was originally translated into Mandarin by a team of psychiatrists in Shanghai Mental Health Centre, with the translation reviewed and modified by members of the CONVERGE team.

The study protocol was approved centrally by the Ethical Review Board of Oxford University (Oxford Tropical Research Ethics Committee) and the ethics committees of all participating hospitals in China. All interviewers were mental health professionals who are well able to judge decisional capacity. The study posed minimal risk (an interview and saliva sample). All participants provided their written informed consent.

2.2.1.2 DNA sequencing and read mapping

Researchers at the Beijing Genomics Institute (BGI) extracted DNA from saliva samples using the Oragene protocol. A barcoded library was constructed for each sample. Sequencing reads obtained from Illumina Hiseq machines were aligned to Genome Reference Consortium Human Build 37 patch release 5 (GRCh37.p5) with Stampy

(v1.0.17) (Lunter and Goodson, 2011) using default parameters after filtering out reads containing adaptor sequencing or consisting of more than 50% poor quality (base quality <= 5) bases. Samtools (v0.1.18) (H. Li et al., 2009) was used to index the alignments in BAM format (H. Li et al., 2009) and Picardtools (v1.62) was used to mark PCR duplicates for downstream filtering.

2.2.1.3 Genotype likelihood calculation

Genotype likelihoods (GLs) were calculated at selected sites using a sample-specific binomial mixture model implemented in SNPTools (Wang et al., 2013). In rounds 1 and 2 we used a pre-release version of SNPTools. Song Li of the BGI used the release version of SNPTools v1.0 in rounds 3 and 4.

2.2.1.4 Genotype accuracy assessment

Genotyping accuracy was assessed by mean per-sample aggregate r-square. This measure was calculated by squared Pearson correlation coefficient (R^2) between imputed dose estimates and test genotypes across all sites in each of a set of allele frequency bins. Allele frequencies of all SNPs were taken from 1000GP1 Chinese haplotypes. However, the test genotypes available changed across rounds. The assessments of rounds 1 and 2 use minor allele frequency (MAF) to bin SNPs, whereas the assessment of rounds 3 and 4 use non-reference allele frequency. When binning SNPs by MAF all dose estimates and genotypes at sites with a non-reference allele frequency > 0.5 were mirrored at 1 in order to avoid inflating estimates of variance in low MAF bins.

In round 1, no genotyping data was available for the CONVERGE samples, so the mapped reads of three 1000GP1 Chinese samples that had been resequenced by the BGI to 10x were downsampled to 1x coverage and processed along with the round 1 CONVERGE samples (see Figure 2.1). Genotypes were called from the

Chr	Start	End	SNPs
20	7000	9000	28247
20	35000	37000	25290
20	57400	59400	28649

(a) 2 Mb regions

Chr	Start	End	SNPs
20	7650	8350	9755
20	35650	36350	9356
20	58050	58750	10329

(b) 700 kb regions

Table 2.1: *Coordinates and number of sites contained in regions used in BEAGLE optimization experiments.* Coordinates are given in kb.

10x data at all sites polymorphic in 1000GP1 ASN haplotypes using The GATK's UnifiedGenotyper. For round 1 these genotype calls were used as the test set for genotype accuracy assessments.

For round 2, 16 randomly selected CONVERGE samples (8 cases and 8 controls) were genotyped on a Zhonghua-8 900k SNP chip, and these genotypes were used for genotype accuracy assessment.

By the time the final call set was available (rounds 3 and 4), a further 56 CONVERGE samples had been genotyped on the Zhonghua-8 SNP chip for a total of 72 samples. Furthermore, 8 CONVERGE samples had been sequenced to 10x coverage on an Illumina platform. Genotypes in these samples were called at SNPs polymorphic in 1000GP1 ASN haplotypes using The GATK's UnifiedGenotyper.

2.2.1.5 Round 1

GLs were calculated from aligned reads of 5,680 CONVERGE samples and three 1000GP1 Chinese test samples downsampled to 1x from aligned reads resequenced to 10x coverage. However, GLs were only calculated at 1000GP1 SNPs in three randomly selected non-overlapping 2 mega base (Mb) regions (chunks) of chromosome 20. The regions are listed in Table 2.1a, and the total number of SNPs in these regions is 82,186.

Genotype probabilities were estimated from the GLs using BEAGLE v3.3.2 and no further command line arguments. BEAGLE was then re-run on the GLs while

using 496 ASN haplotypes as a reference panel. Finally, BEAGLE was rerun using these haplotypes as a reference panel and with twice as many iterations (20) as is the default (10). Genotyping accuracy was assessed as described in subsubsection 2.2.1.4. The results of this assessment can be seen in Figure 2.4.

A set of three 700 kilo base (kb) regions listed in Table 2.1b was created in the center of each of the three 2 Mb regions used so far. Genotype probabilities were estimated from GLs in the 700 Kb regions using BEAGLE together with the ASN reference panel used so far, and using 10 (the default), 6, and 4 iterations. Genotyping accuracy was again assessed as described in subsubsection 2.2.1.4, and the results of this assessment can be seen in Figure 2.5. These data were also assessed as a function of 700 kb chunk location relative to the start of each 700 kb chunk (see Figure 2.6).

Based on the results of these experiments, for subsequent rounds we chose to run BEAGLE for six iterations on 700 kb chunks overlapping by 50 kb.

2.2.1.6 Round 2

New CONVERGE samples were made available by the BGI, bringing the total number of samples in the analysis data set to 9,300. These samples were a subset of the 11,670 samples recruited for CONVERGE. Genotyping data of 16 CONVERGE samples from a Zhonghua-8 SNP chip were also made available by the BGI. This allowed the removal of the reads of three downsampled 1000GP1 samples from the analysis data. It was discovered that 70 East Asian 1000GP1 haplotypes had been missed when creating the ASN reference panel. These haplotypes were added together with the six haplotypes of the previous test samples to the ASN reference panel. This increased the number of haplotypes in ASN from 496 to 572.

Following best practices (DePristo et al., 2011; The 1000 Genomes Project Consortium, 2012), a base quality score recalibration table was created for each BAM file using The GATK (v2.3.9), and a reference SNP and indel list from dbSNP version

137 excluding all sites after version 129. For each BAM file, GATKlite (v2.2.15) was used to remove the reads that were not properly mapped, and to recalibrate base quality scores from the recalibration tables (BQSR[2]). This involved removing 1-5% of reads per sample.

To investigate the benefit of using BQSR, BQSR was performed on ten randomly selected CONVERGE BAM files for the whole genome and only on chromosome 20. How the base qualities are adjusted by base quality score was extracted from the resulting recalibration files (Figure 2.7).

In order to save on run time, we removed about 60% of 1000GP1 SNPs that were monomorphic in ASN haplotypes, leaving 13,837,179 SNPs at which we performed GL calculation and genotype imputation. The results of genotype imputation accuracy assessment of the round 2 genotype doses across the whole genome are given in Figure 2.8.

Replacing BEAGLE with a novel method presented in chapter 3 was also investigated. However, the results presented in subsection 3.2.1 indicate that BEAGLE is both faster and more accurate than my method on the round 2 data, so BEAGLE was used for rounds 3 and 4.

2.2.1.7 Rounds 3 and 4

The work in this paragraph was performed by Na Cai of the Wellcome Trust Centre for Human Genetics. Variant discovery and genotyping at all polymorphic SNPs in the ASN reference panel was performed simultaneously using post-BQSR sequencing reads from all samples using the GATK's UnifiedGenotyper (version 2.7-2-g6bda569).

[2]Base quality score recalibration (BQSR) is an empirical method for recalibrating aligned sequencing read base qualities (DePristo et al., 2011). The method masks out all sites on a provided list of known SNPs (for example from 1000 Genomes), and assumes that all other non-reference bases are base calling errors. All bases are categorized by their reported quality score, their relative position on the read, and their dinucleotide context. For each category an empirical error rate is calculated. An adjustment is calculated for every category such that categories have the error rate that they have empirically, and every base quality is adjusted by each applicable category adjustment.

Na Cai set the option `--genotype_likelihood_model` to `BOTH`, used default annotation outputs for variant calls, and set the `--dbSNP` option in order to use dbSNP v137 rsids to fill in the variant ID column of the output variant call format (VCF) files. Variant quality score recalibration was then performed on these sites using the GATK's VariantRecalibrator (version 2.7-2-g6bda569) and the bi-allelic SNPs from ASN samples as a true positive set of variants. A sensitivity threshold of 90% to SNPs in the ASN panel was applied for SNP selection for imputation after optimizing for Transition to Transversion (TiTv) ratios in SNPs called. This gave a total of 21,356,798 (9,053,391 known in the ASN panel and 11,486,024 novel) bi-allelic SNPs identified from all chromosomes and unassembled contigs. Na Cai excluded all sites with a minor allele count of 0 in the genotypes generated with the GATK's UnifiedGenotyper, and put forth 20,539,441 SNPs from the autosomes and chromosome X for imputation of genotype probabilities.

My pipeline was sent to the BGI, where it was run by Song Li. Song Li called genotypes from GLs using BEAGLE (version 3.3.2). For sites that were polymorphic in the ASN haplotypes, the BGI ran BEAGLE for six iterations using the 572 ASN haplotypes as a reference panel as part of round 3. For all SNPs discovered in the CONVERGE data set, the BGI ran BEAGLE using ten iterations without a reference panel as part of round 4. In both cases, the BGI ran BEAGLE in parallel on chunks containing roughly 3000 SNPs, such that neighboring chunks shared 600 SNPs. This corresponds to approximately 700 kilo base windows with 50 kilo base overlaps. Song Li created the final genotype call set by replacing genotypes at sites in round 4 that were also called in round 3 by the genotypes called in round 3. Figure 2.9 is the result of my assessment of the quality of the round 3 and round 4 genotypes, which is described in subsubsection 2.2.1.4.

2.2.2 Results

2.2.2.1 Round 1

The first goal of round 1 was to prototype a pipeline for calling genotypes from aligned 1x coverage reads of 5,680 CONVERGE samples that would scale to the full CONVERGE data set. Following the 1000GP1 paper (The 1000 Genomes Project Consortium, 2012), SNPTools was used to call GLs. We did not perform site discovery, but instead followed Pasaniuc et al. (2012) in calling GLs at all 1000GP1 SNPs. Initially, we tried running BEAGLE on the whole of chromosome 20 as described by Pasaniuc et al. (2012), but this required more memory than was available on our servers. We opted instead to run BEAGLE in parallel on 2 mega base (Mb) regions. For the purpose of prototyping in round 1 we only worked with three randomly chosen 2 Mb regions (chunks) of chromosome 20 listed in Table 2.1a. BEAGLE memory usage increases linearly with the number of sites imputed. In anticipation of running on our local cluster, we needed to choose a chunk size that would run on the high memory nodes, which at the time had 8 giga bytes (GB) of memory available per job. The largest chunk size that BEAGLE would consistently run on using 8 GB of memory was 2 Mb.

In an effort to increase BEAGLE imputation accuracy, we examined the effect of using the ASN reference panel on genotype imputation accuracy as shown in Figure 2.4. It was found that using the panel increased R^2 by 20 and 10 percentage points at minor allele frequencies (MAF) $< 5\%$ and $> 10\%$, respectively. This matched Pasaniuc et al. (2012) findings that using a reference panel substantially increases genotype imputation accuracy using BEAGLE for sparse ($<$2x) and extremely low (0.1x) coverage sequencing data. It was further explored if doubling BEAGLE iterations from the default 10 to 20 provides any benefit, but Figure 2.4 seems to indicate that this is not the case.

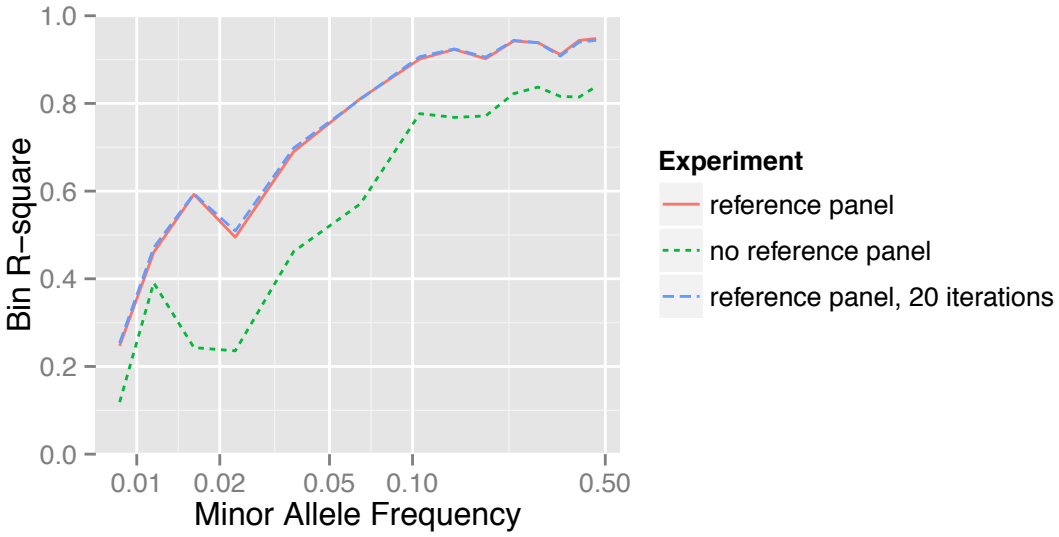

Figure 2.4: *R^2 between BEAGLE imputed genotype dose estimates and genotypes in three downsampled Chinese 1000GP1 test samples.* Genotypes were called from 10x Illumina sequencing data at 1000GP1 SNPs in three 2 Mb regions of chromosome 20. SNPs were binned as a function of MAF in 1000GP1 Chinese. BEAGLE was run for 10 iterations with and without the ASN reference panel (496 haplotypes) and for 20 iterations with this reference panel. The lines represent the mean R^2 of the three samples in each bin.

During round 1, informal timing runs of BEAGLE indicated that BEAGLE was likely going to require a large amount of compute time. In response, we investigated ways to reduce BEAGLE run time by reducing BEAGLE sampling iterations, and switching to developing the pipeline for the more numerous compute nodes of our cluster. These nodes only have 2.67 GB of memory per job, and this meant that we needed to further reduce chunk size. From past experience of running BEAGLE on sequencing data, we estimated that a chunk size of 700 kb would be sufficiently small to fit the memory constraint, and this was in fact the case. From Figure 2.5 it appears that genotypes imputed in 700 kb chunks perform as well as genotypes imputed in 2 Mb chunks when using BEAGLE's default number of 10 iterations. The figure also shows that although imputation accuracy for four iterations compared to more iterations does decrease appreciably, imputation accuracy for six and ten iterations is similar. Based on this data we chose to run BEAGLE with six iterations for round 2.

BEAGLE is an LD-based algorithm. As such, edge effects leading to reduced imputation accuracy are a concern. In two of the three 700 kb chunks examined, there appeared to be a small drop in imputation accuracy in the last 50 kb of the chunks compared to the genotypes imputed from overlapping 2 Mb chunks (data not shown). The average of the three chunks is displayed in Figure 2.6. To be safe, we opted to design the round 2 chunks such that they include an extra 50 kb on either side that overlaps with neighboring chunks. These extra 50kb on either side are dropped when the chunks are merged after imputation with BEAGLE.

We also investigated increasing the reference panel from the 496 ASN haplotypes to almost all 1000GP1 haplotypes (2178). However, we found that this increased BEAGLE resident memory requirements from well under 2 GB to well over 4 GB and approximately tripled run time. Together, this would have increased run time nine fold on the compute nodes of our cluster, which was an unacceptable cost. As a result, we did not further investigate imputation accuracy using the complete

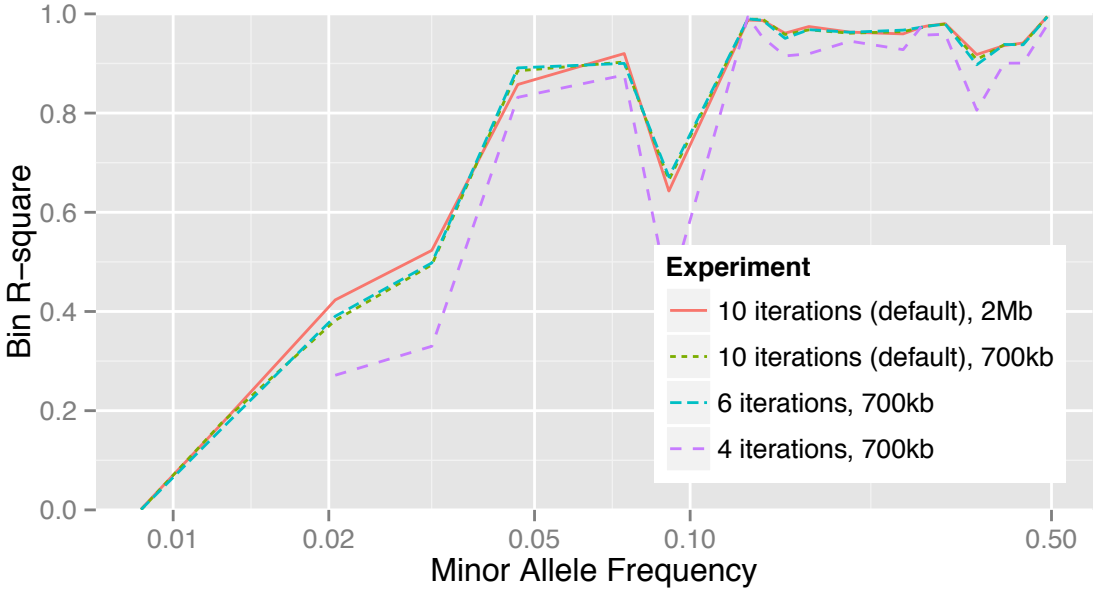

Figure 2.5: *R^2 of BEAGLE imputed dose estimates decreases at four BEAGLE iterations and remains high in 700kb chunks.* Genotypes were called from 10x Illumina sequencing data. SNPs were binned as a function of MAF in 1000GP1 Chinese. The 700kb regions used are listed in Table 2.1b. BEAGLE was run for four, six, and ten iterations, and for ten iterations on the 2 Mb regions listed in Table 2.1a. The 2 Mb region data not overlapping with 700 kb regions were removed The lines represent the mean R^2 of three Chinese 1000GP1 samples at each bin. All BEAGLE runs used the ASN reference panel (496 haplotypes).

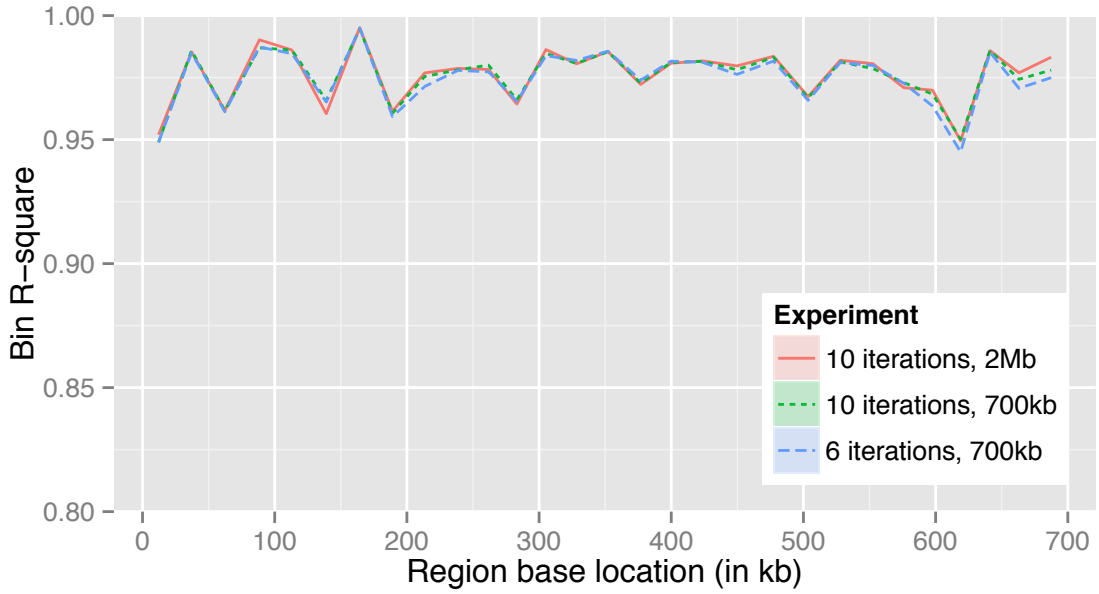

Figure 2.6: *Mean R^2 of three 700kb regions imputed with BEAGLE compared to the same location within 2 Mb regions.* Test genotypes were called from 10x Illumina sequencing data. BEAGLE was run for 6 and 10 iterations on three 700 kb regions listed in Table 2.1a, and for 10 iterations on three 2 Mb regions listed in Table 2.1a. In all cases, BEAGLE was run with the ASN reference panel (496 haplotypes). SNPs were binned as a function of relative region location from the start of each 700 kb region. The 2 Mb region data not overlapping with 700 kb regions were removed. The lines represent the mean of three region means of three Chinese 1000GP1 samples at each bin.

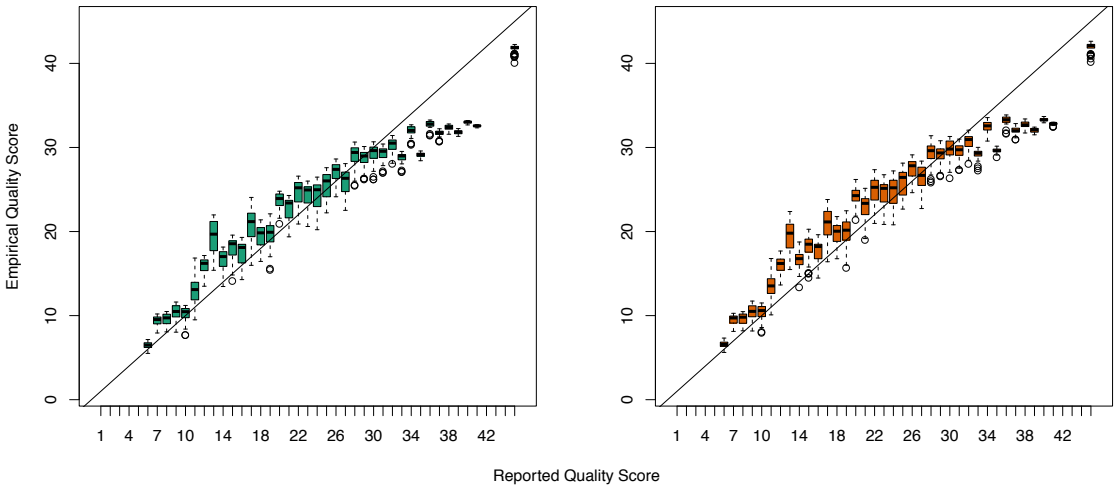

Figure 2.7: *Empirical versus reported quality scores of 10 randomly selected CONVERGE BAM files.* This is a diagnostic plot created by the BQSR procedure. The 1000GP1 SNPs were used as true variant sites. Quality scores are Phred scaled. Some inflation in reported high quality scores is evident. **Left**: Whole genome. **Right**: Chromosome 20.

1000GP1 haplotype reference panel.

2.2.2.2 Round 2

From running BQSR on ten randomly select CONVERGE BAM Files (Figure 2.7) it appears that the CONVERGE base quality scores are not perfectly calibrated, and may benefit from recalibration. A more thorough assessment of the effect of BQSR on GL calculation and genotype calling with BEAGLE was not performed.

Whole-genome imputation of 9,300 CONVERGE samples at 13,837,179 SNPs using 572 reference haplotypes resulted in a mean R^2 of 92% for SNPs with a MAF $>$ 5% (see Figure 2.8). This means that an association study on the imputed genotypes would have the power to detect an association at alleles with MAF $> 5\%$ equivalent to the power of approximately 8,556 samples genotyped on a microarray chip (Pritchard and Przeworski, 2001).

Unfortunately we did not keep track of the amount of compute used to impute all genotypes with BEAGLE. However, from subsequent experiments on five randomly

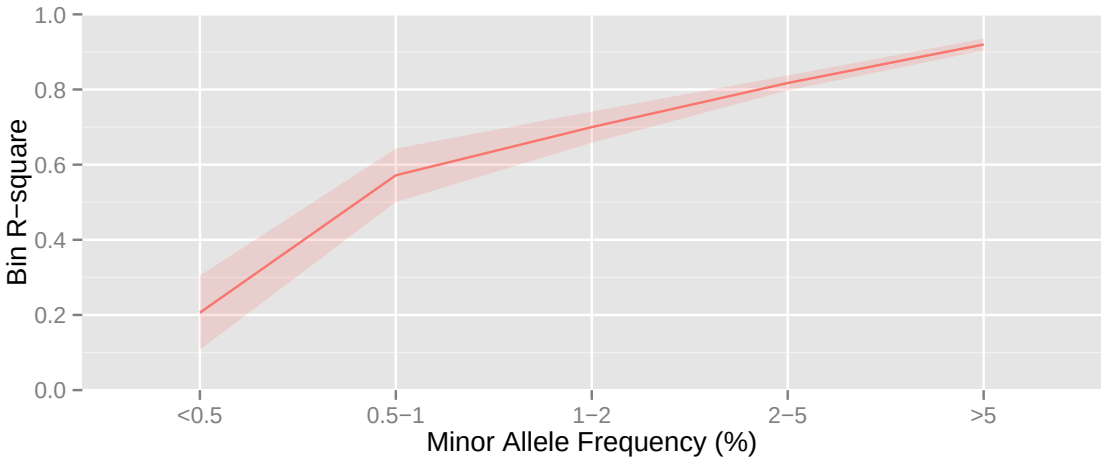

Figure 2.8: *R^2 between round 2 imputed dose estimates and genotypes in 16 CONVERGE samples across the whole genome.* Genotypes were called from a Zhonghua-8 900k SNP chip and binned by minor allele frequency. The solid line and semi-transparent ribbon represent the mean and two times the standard deviation of R^2 around the mean, respectively, of the 16 samples at each bin.

selected 1024 site chunks on chromosome 20, we estimate that the cost of computation was about 1,560 CPU days. This was after reducing the number of sites to impute by 60% from all 1000GP1 SNPs to only the ones polymorphic in ASN haplotypes. The methods of this analysis are described in section 3.2.

Imputation accuracy in round 2 was as good as, or better than round 1 imputation accuracy and it matches what has been predicted for 1x coverage data from simulation studies (Pasaniuc et al., 2012, Figure 1). However, the increase in accuracy from round 1 to round 2 is confounded by the technology used to determine "true" genotypes. In round 1, genotypes were called from 10x Illumina sequencing data at all 1000GP1 SNPs (approximately 38 million SNPs), while in round 2 genotypes were called from a genotyping array at only 890,371 SNPs. It is possible that the sites used in the SNP chip may be biased towards sites that are easy to genotype because they are in more accessible parts of the genome. Therefore, these sites may on average have higher coverage than the rest of the 1000GP1 sites. Evidence in a disparity of genotyping accuracy between SNP chip and 10x sequencing test set data is shown in Figure 2.9.

2.2.2.3 Rounds 3 and 4

The genotyping accuracy on chromosome 20 of the final call set (Figure 2.9) is substantially improved over the whole genome assessment of round 2 (Figure 2.8). However the chromosome 20 analysis does match a similar analysis performed by Song Li on the whole genome in CONVERGE Consortium (2015, supplemental Table 2). A small part of the difference between round 2 and the final genotype call set imputation accuracy will be due to increased sample size in the final call set. However, during the implementation of my pipeline at the BGI, we discovered that a bug had been found in the GL caller of SNPTools between the pre-release version we had been using and the release version (1.0). This bug was causing base qualities to be converted from Phred scale incorrectly. Instead of using the formula

$$P = 10^{-Q/10}, \tag{2.1}$$

the pre-release version of SNPTools was using the incorrect formula

$$P = e^{-Q/10}, \tag{2.2}$$

where P is the probability of a base being miscalled, and Q is the Phred scaled base quality. This bug caused the pre-release version of SNPTools to interpret base qualities as having more errors than they actually did. This in turn would have led to an artificial flattening of already quite flat GLs, which leads to reduced genotype imputation quality. This may explain why imputation accuracy is substantially improved in the final genotype call set over the round 2 call set.

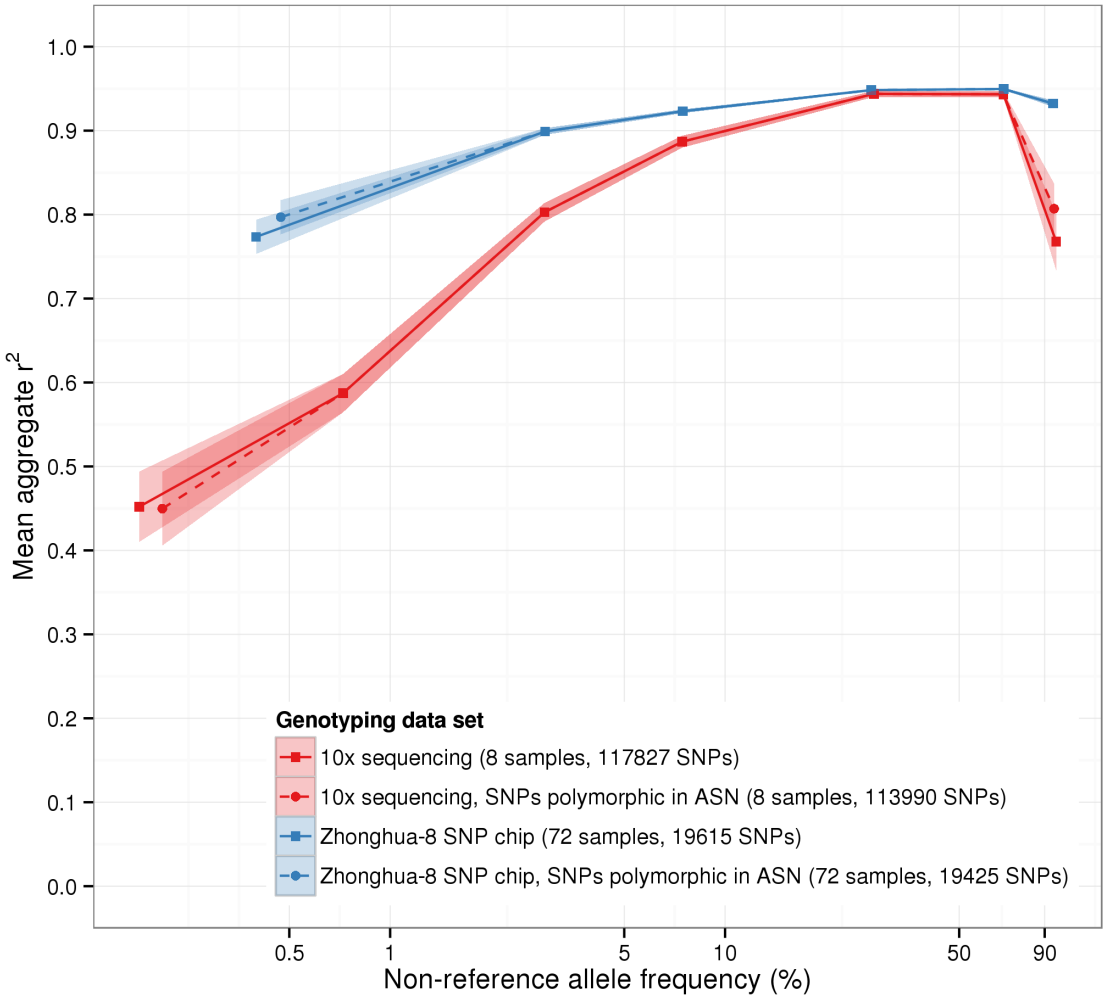

Figure 2.9: *R^2 between final genotype calls (round 3 + round 4) and validation genotypes on chromosome 20.* Validation genotypes were called from 10x Illumina sequencing and Zhonghua-8 900k SNP chip data. SNPs were binned according to their non-reference allele frequency in 1000GP1 Chinese haplotypes. Lines and semi-transparent ribbons respectively represent the mean and two times the standard deviation of R^2 around the mean of the samples at each bin. "SNPs polymorphic in ASN" are SNPs that were part of round 3.

2.3 Haplotype calling

Although BEAGLE provides haplotype estimates from genotype imputation and phasing, the merging of the round 3 and round 4 phased genotype call sets means that the final genotype call set consists of two interleaved haplotype sets, where SNPs that were only seen in round 4 are randomly phased with regards to the SNPs selected for round 3. An approach that is easy and has been used successfully before (see Huang et al. (2015) and subsection 4.2.6) is to rephase these haplotypes using SHAPEIT. In this case SHAPEIT v2.r790 was used. Genetic maps were obtained from the IMPUTE2 website. Chromosomes 13 - 22 and X were phased using 12 threads and default parameters. Chromosomes 1-12 were phased in four chunks that overlap by 1 Mb. The phased chunks were ligated together using ligateHAPLOTYPES, which can be found on the SHAPEIT2 website.

2.3.1 Haplotype downstream imputation accuracy assessment

We investigated the quality of the CONVERGE haplotypes by performing imputation of genotypes from CONVERGE haplotypes into a pseudo-GWAS panel consisting of samples sequenced to high coverage by Complete Genomics (CG) at sites on the Zhonghua-8 SNP chip. All other sites of the CG sequencing data were masked out and used to determine downstream genotype imputation accuracy of the CONVERGE haplotypes.

2.3.1.1 Complete Genomics data preparation

Complete Genomics (CG) whole genome genotype calls from high coverage sequencing for 10 CHS samples were obtained from the 1000 Genomes Project website in VCF format. All sites that are not bi-allelic SNPs were filtered out.

The sample names of Complete Genomics samples used for downstream imputa-

tion accuracy assessment are: HG00448, HG00436, HG00663, HG00628, HG00692, HG00592, HG00683, HG00437, HG00690, HG00684.

A pseudo-GWAS panel was created by filtering out genotypes at all sites that did not have positions in the `Zhonghua-8v1-1_a` positions list. Genotypes at the filtered out sites were kept to create the validation data set if those sites had no missing genotypes in the 10 CHS samples.

2.3.1.2 1000 Genomes Phase 3 reference panel preparation

The 1000 Genomes Phase 3 version 5 (1000GP3) haplotypes were obtained from the 1000 Genomes FTP server. All sites that had missing genotypes, unphased genotypes, or were not SNPs were removed. Multi-allelic SNPs were broken into bi-allelic SNPs. The 10 CHS Complete Genomics samples used for the validation set were removed.

2.3.1.3 Genotype imputation into pseudo-GWAS panel

The 1000GP3 and CONVERGE haplotypes were used to impute genotypes into the pseudo-GWAS panel using IMPUTE2 with the non-default options `-k_hap 1000 -Ne 15000`. Imputation was performed in 2-6 Mb chunks. Although the IMPUTE2 website recommends 5mb chunks, 2 Mb chunks were used in order to reduce IMPUTE2 memory usage. Some 2 Mb chunks did not contain enough GWAS panel SNPs, so those chunks merged with their neighbors until they contained enough SNPs.

2.3.1.4 Genotype imputation accuracy

Genotype imputation accuracy into the pseudo-GWAS set was evaluated across all autosomes at the intersection of sites in the CONVERGE and 1000GP3 panels and the validation data. Sites were binned by non-reference allele frequency derived from the 1000GP3 East Asian haplotypes and the CONVERGE haplotypes. R^2 was calculated in aggregate across all sites and all samples for each allele frequency bin. The results

of the assessment can be found in Figure 2.10. The reference panels compared in this figure are the 1000 Genomes Phase 3 reference panel, the CONVERGE haplotypes, a combined panel consisting of 1000 Genomes Phase 3 and CONVERGE haplotypes, with missing genotypes set to the homozygous major allele. The CONVERGE panel was also imputed using a larger number (2,000) for the IMPUTE2 parameter `-k_hap`.

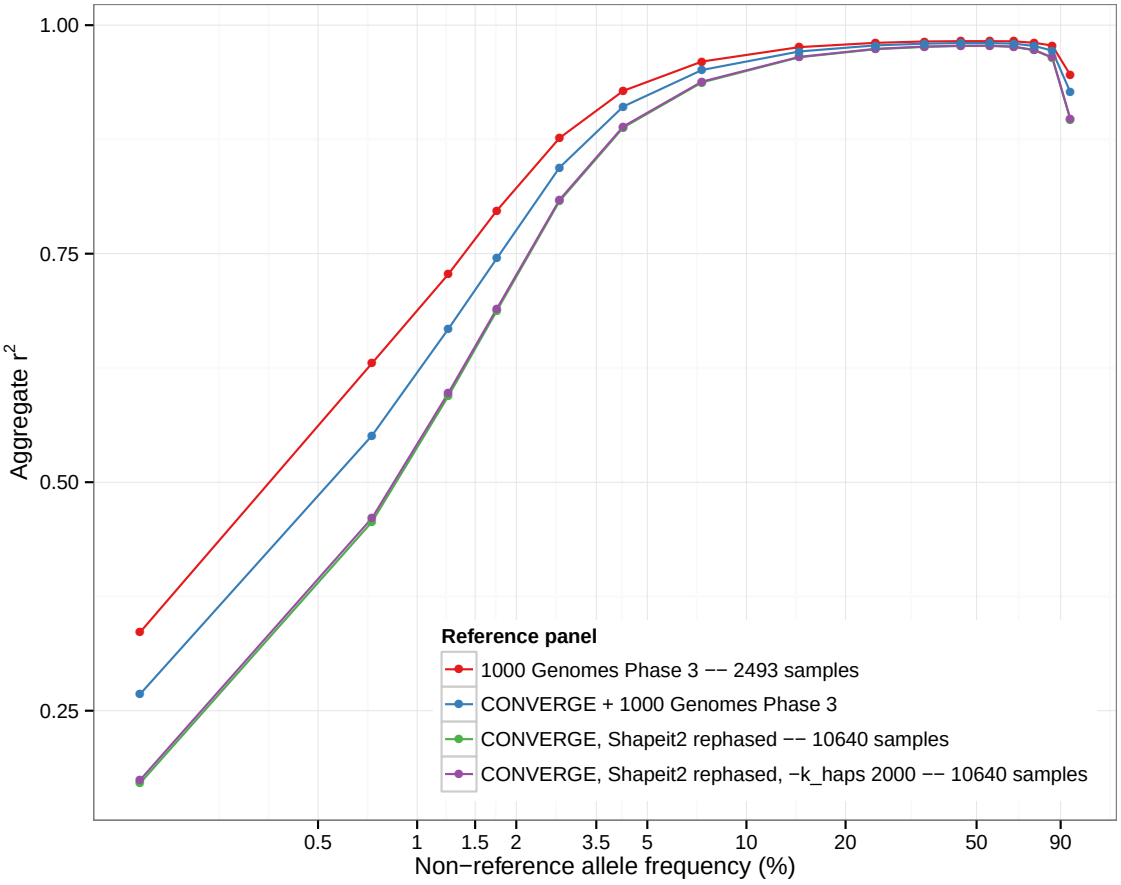

Figure 2.10: *Downstream imputation accuracy of imputation into 10 CHS samples on chromosomes 1-22 using 1000 Genomes Phase 3 and CONVERGE reference panels.* Chip sites: 868,387 Zhonghua-8 SNPs. R^2 calculated at 8,400,163 SNPs polymorphic in the 10 CHS samples.

2.4 Discussion

In this chapter the work that I and some of the CONVERGE consortium researchers have done to develop a pipeline to calculate genotype probabilities and haplotypes of

11,670 CONVERGE samples sequenced to 1.7x coverage is described. The rephasing of the CONVERGE haplotypes is also described, along with the testing of their suitability as a reference panel for genotype imputation into a Zhonghua-8 SNP chip.

My work shows that it is possible to use 1.7x sequencing data to genotype samples for a GWAS. In fact, a GWAS using the CONVERGE data was successful in detecting and replicating the first two hits for MDD (CONVERGE Consortium, 2015).

Delaneau and Marchini (2014) assessed imputation accuracy in ten European and four South Asian samples for which high coverage sequencing data from Complete Genomics was available. Imputation accuracies measured by aggregate Pearson R^2 in the alternate allele frequency bin of 1%-2% were 0.73 and 0.70, in the European and South Asian populations, respectively. At the allele frequency of 50% an R^2 of 98% and 95% were measured in the European and South Asian samples, respectively. The imputation accuracy measured by R^2 in Figure 2.9 in the 1%-2% and 50% allele frequency bins is about 84% and 94%, respectively. This represents a substantial increase in genotype imputation accuracy at the first allele frequency bin and a small or no decrease at the second allele frequency bin. Based on this comparison it is possible that the CONVERGE approach to calling genotypes from sparse sequencing data may outperform imputation from 1000 Genomes Phase 1 haplotypes into an Illumina 1M SNP chip using IMPUTE2 for the purpose of GWAS.

It is also worth comparing the cost of sequence-based and SNP chip-based genotype imputation. On January 7th, 2016, the cost of 150 base pair paired-end sequencing to 2x coverage of 11,670 human samples (the size of the CONVERGE data set) on a HiSeq4000 at the Oxford Genomics Centre was £ 176 per sample (£ 60 library prep, and £ 116 sequencing reagents), which is £ 2,053,920 in total. In contrast, the cost of genotyping 384 samples on a Zhonghua-8 v1.2 Illumina bead chip was £ 29,433 (£ 77 per sample) on January 5th, 2016, according to the Illumina website (`www.illumina.com`), which is about £ 898,590 for 11,670 samples.

The computational cost of imputing all 1000GP1 SNPs into Wellcome Trust Case Control Consortium (The Wellcome Trust Case Control Consortium, 2007) SNP chip data was measured by B. Howie, Fuchsberger, et al. (2012) as 49 minutes using 762 European haplotypes together with a pre-phasing strategy that uses SHAPEIT2 to pre-phase the SNP chip data and IMPUTE2 to impute genotypes into the pre-phased haplotypes. This amounts to a cost of 405 CPU days to impute genotypes into the autosomes of 11,670 samples, assuming that the SHAPEIT2 plus IMPUTE2 strategy grows roughly linearly in run time with sample size. In contrast, the cost of genotype imputation of the CONVERGE data set with BEAGLE into all 1000GP1 SNPs would be about 3,900 CPU days, which would be the bulk of the computation necessary to run my pipeline. 405 and 3,900 CPU days translate to a cost of 580 USD and 5,593 USD, respectively, on a compute optimized Amazon cloud server in Ireland (`aws.amazon.com/ec2/pricing`, server name: c3.8xlarge at 1.912 USD/hour) on January 5th, 2016.

The cost of storing 35 TB, roughly equivalent to 12,000 BAM files at 1.7x coverage, on the Amazon S3 cloud storage service in Ireland (`aws.amazon.com/s3/pricing`, Standard - Infrequent Access Storage) was 448 USD per month on January 5th, 2016. The SNP chip data is negligible in cost to store.

The costs of data generation, computation, and data storage are all smaller for a SNP chip-based genotype imputation than a sequencing-based genotype imputation strategy. Although the past decade has seen a tremendous decrease in sequencing costs, and it is expected that sequencing costs will further decrease in the future (Wetterstrand, 2013), at this time, for a sample size comparable to CONVERGE, a SNP chip-based GWAS is likely still the better choice of strategy. As of the writing of this thesis, the second phase of CONVERGE has been funded. Unsurprisingly, the phase 2 samples will be genotyped using a SNP chip array.

For other sample sizes computational cost will vary substantially, because BEA-

GLE scales quadratically with sample size (see Figure 3.3). This means that it will be substantially cheaper to impute genotypes in a data set consisting of fewer samples, and substantially more expensive to impute genotypes of a data set with a larger sample size. One approach that may be feasible in this case is to use BEAGLE to impute genotypes in small batches of randomly selected samples. It may also be worth investigating if a novel method presented in subsection 3.2.1 is appropriate at other sample sizes.

Another trend to consider is that haplotype reference panels are expected to increase in size (Y. Li, Willer, et al., 2010). For example, in chapter 4 work on imputing genotypes in over 30,000 samples for phasing as part of the Haplotype Reference Consortium (HRC) is presented. From my limited experimentation in round 2, it appears that BEAGLE also scales quite poorly with reference panel size both in run time and in memory usage. However, it may be worth investigating if a larger population-specific reference panel brings with it a substantial boost in genotype imputation accuracy.

Figure 2.10 shows us that even though the CONVERGE panel is four times as large as the 1000GP3 panel, imputation accuracy using the CONVERGE panel is substantially reduced compared to imputation accuracy from the 1000GP3 panel. This is likely due to the 1000GP3 panel consisting of higher quality genotypes than the CONVERGE panel, which would lead to higher quality haplotypes. It appears that the CONVERGE approach is not suited for creating high quality reference haplotypes.

Pasaniuc et al. (2012) argue that sequencing at coverage as low as 0.1x might one day deliver a cheaper alternative to microarray genotyping. If the cost of 0.1x sequencing does indeed come down below the cost of SNP chip-based genotyping, then perhaps this may be true. However, the costs of computation and storage of BAM files still remain as substantial hurdles to a move from SNP chip-based to sequencing-based genotyping.

Chapter 3

A new method for genotype calling and phasing for large scale low depth sequencing studies

Over the past few years, short read sequencing has become cheap enough to give rise to collections of whole genome, low coverage sequencing data of over ten thousand human samples (see CONVERGE Consortium (2015) and McCarthy et al. (2015)). Current best practice for calling genotypes from low coverage sequencing data is to call genotype likelihoods (GLs) at a set of previously identified or newly discovered sites, and then to use BEAGLE (B. L. Browning and Yu, 2009; S. R. Browning and B. L. Browning, 2007) to obtain phased genotype calls from GLs (CONVERGE Consortium, 2015; Genome of the Netherlands Consortium, 2014; The 1000 Genomes Project Consortium, 2012).

The run time of BEAGLE scales super-linearly with sample size (B. L. Browning and S. R. Browning, 2007; Loh, Palamara, and Price, 2015). While the CONVERGE Consortium (2015) manages to use BEAGLE to call genotypes on just over 10k samples using a cluster and several weeks of time, at the sample size of the HRC (32,611

samples, chapter 4), BEAGLE is not a viable approach anymore.

The Marchini Group's experience from the 1000 Genomes Project Phase 1 (The 1000 Genomes Project Consortium, 2012) is that the SNPTools (Wang et al., 2013) method outperformed its competitor Thunder (Y. Li, Willer, et al., 2010) in genotype calling accuracy on the 1000 Genomes Phase 1 data set. The SNPTools method for calling phased genotypes from GLs is similar to the Thunder approach. A sample is modeled as an imperfect mosaic of a set of haplotypes drawn from the current haplotype estimates of the remaining samples using a hidden Markov model (HMM) according to a N. Li and Stephens (2003) model. Both methods use a Markov Chain Monte Carlo (MCMC) scheme to iteratively update the haplotypes of one sample at a time based on the current estimate of the haplotypes of all the other samples. In the case of SNPTools, the MCMC scheme is a Simulated Annealing Metropolis-Hastings (MH) sampler (Kirkpatrick, Gelatt, and Vecchi, 1983) that is iterated $2N$ times, where N is the number of samples being phased. The most striking difference between the two methods is that to update a sample's haplotypes, SNPTools constrains itself to copying from only four haplotypes, while Thunder uses a much larger random subset of the haplotypes. Both SNPTools and Thunder outperform BEAGLE in genotype and haplotype imputation accuracy (Y. Li, Willer, et al., 2010; Wang et al., 2013), albeit at a higher computational cost.

The super-linear complexity of BEAGLE and $O(N^2)$ complexity of SNPTools makes application of these models to upwards of 10k samples intractable. There is no method for calling genotypes from GLs that has better scaling properties.

The aim of this chapter is to adapt the SNPTools method for genotype calling from GLs in a way that scales better with sample size than BEAGLE, while retaining similar genotype calling and haplotype phasing accuracy. We approach this aim in three parts: First, a mathematical description of the SNPTools model as described by Wang et al. (2013) is presented, but augmented with details defined by their

implementation. Second, we critique the implementation's deviations from theory and investigate how correcting these deviations affects haplotype phasing accuracy. Third, modifications to the SNPTools algorithm that greatly speed up genotype calling from GLs are described and benchmarked:

- a modification to use haplotype reference panels of samples not being phased
- a modification to incorporate pre-existing haplotypes for the samples being phased
- a modification to allow running the SNPTools algorithm in parallel on a graphics processing unit (GPU)

These modifications are available as a binary called GLPhase.

3.1 Background: The SNPTools genotype calling and phasing model

Although Wang et al. (2013) only describe their SNPTools genotype calling and phasing model in a general form, they did make their source code publicly available. The following mathematical description of the SNPTools genotype calling and phasing model has been created from information in Wang et al. (2013) where possible, but in large part from the publicly available source code and standard literature on HMMs (Cawley and Pachter, 2003; Durbin et al., 1998; Rabiner, 1989). At times the source code diverges from models described in the standard literature. First, the SNPTools model is described in a manner consistent with the standard literature. Later, the differences between the SNPTools source code and the literature are examined.

In the remainder of this chapter we will refer to the SNPTools genotype calling and phasing model, as implemented by Wang et al. (2013) in the C++ source code file `impute.cpp` that can be downloaded from `www.http://sourceforge.net/projects/`

`snptools` as part of the SNPTools version 1.0 package, as "the SNPTools model" when referring to the model, and as "SNPTools" when referring to the implementation of that model.

3.1.1 Notation

In the SNPTools model we would like to estimate the haplotypes of N individuals at L bi-allelic SNPs in a region (or chunk). The model uses a pseudo-Gibbs sampling scheme where each individual's haplotypes are estimated in parallel conditional on current estimates of a subset of all other haplotypes. Therefore, it is sufficient to describe a single individual's update. We assume that an individual's latent diplotype Y can be expressed as a vector of ordered pairs of alleles $Y = Y_1, \ldots, Y_L$, where $Y_l \in \{(0,0),(0,1),(1,0),(1,1)\}$ represents the individual's pair of alleles at the l^{th} SNP. That is, Y_{l1} is the allele from parent 1 and Y_{l2} is the allele from parent 2 at the l^{th} SNP. The possible alleles at the SNP are arbitrarily coded as 0 and 1, and the individual's genotype at SNP l i's simply the sum of the alleles of Y_l. For each individual a small number of short reads, R, have been sequenced. At each SNP, the per-sample data supporting each of the possible genotypes has been quantified in three genotype likelihoods (GLs) that take the form

$$P(R_l|Y_l) = P(R_l|Y_{l1} + Y_{l2} = g), \quad g \in 0,1,2. \tag{3.1}$$

This is the probability of observing the reads R_l at SNP l assuming that the genotype of the individual at that SNP is g.

We will use H to denote the set of all current haplotype estimates, where H_{hl} is the allele of the h^{th} haplotype at the l^{th} SNP, and H_{-n} is the set of haplotypes without the haplotypes corresponding to the n^{th} individual. The number of haplotypes in H is $2N$.

SNPTools samples a new diplotype Y of length L for individual n based on H_S, a subset of H_{-n}, that is consistent with R. $S = (s_1, \ldots, s_4)$, where $s_j \in \{1, \ldots, 2N\}$, is a tuple representing the haplotypes of surrogate parents selected from haplotypes in H_{-n}. The first two haplotype indices of S represent the surrogate father, and the other two represent the surrogate mother.

SNPTools employs an MH sampling scheme (see subsection 3.1.3) to determine a good tuple S through an HMM (see subsection 3.1.4). Part of the HMM is a latent state sequence $Z = (Z_1, \ldots, Z_L)$, with one state for each SNP. Each state Z_l may take the value of one of four pairs contained in the tuple $u = ((1,3),(1,4),(2,3),(2,4))$. u contains all possible pairs of indices into S, reflecting the constraint that any state Z_l must refer to one of each parents' chromosomes. We will be referring to the HMM parameters as $\Lambda = (H_S, r, \mu)$. These parameters are described further in subsection 3.1.4.

3.1.2 Choice of L

The small number of states in the HMM limits the number of recombinations between surrogate parent haplotypes that can be modeled across L sites. Hence, the choice of L has an impact on phasing accuracy. Wang et al. (2013) claim that smaller chunks provide better phasing accuracy than larger chunks. In support of this hypothesis, Wang et al. (2013) compare the genotype imputation accuracy of using 512 sites and 1024 sites per chunk in 1000 Genomes Phase 1 sequencing and SNP genotyping data.

However, the optimal size of L probably depends on the number of samples being phased, the degree of relatedness between the samples, and the genetic distance between the ends of a chunk. The SNPTools HMM can effectively model only one recombination within each parent, so chunk sizes should not be too large. On the other hand, the closer any one sample is related to it's nearest relatives in the sample, the longer one can expect shared regions of IBD between those samples to be. If shared

stretches of IBD are large, then the SNPTools model should have a better chance of detecting samples that share long stretches of IBD if L is large. The longer the stretches of IBD are that SNPTools can detect, the better the phasing and imputation quality of SNPTools should be. For the HRC we also tried chunk sizes of 2048 and 4096 (data not shown), but we saw no difference in downstream imputation accuracy compared to using 1024 sites.

The Wang et al. (2013) implementation of the SNPTools model stores alleles as bit vectors inside unsigned 32 bit unsigned integers. Therefore, the most efficient storage of haplotypes is achieved if L is a multiple of 32. In GLPhase haplotypes are stored in 64 bit unsigned integers, and the most efficient choices of L are multiples of 64.

3.1.3 The SNPTools sampling scheme

Wang et al. (2013) employ a nested sampling scheme with two levels to estimate the haplotypes H. Before any sampling, the alleles of H are randomly set to 0 or 1 with equal probability.

3.1.3.1 Diplotype sampling

A new diplotype H'_n of each individual n is generated conditional on a haplotype subset H_S and the observed data R. All individuals are updated in parallel from different subsets of the same set of haplotypes H. Once all diplotypes in the new haplotype set H' have been updated, H is updated with H' and the scheme starts a new iteration. SNPTools performs M_1 iterations. By default, SNPTools uses $M_1 = 256$. The output of SNPTools is a set of allelic probabilities generated by averaging the haplotype sets of the last 200 (sampling) iterations.

The first 56 M_1 iterations are burn-in iterations in which the MCMC sampler is not yet sampling from a stationary distribution. In the first half of these burn-

in iterations, the Wang et al. (2013) implementation of the SNPTools model also contains a penalty variable

$$p_m = \left(\min \left[\frac{2(m+1)}{b}, 1 \right] \right)^2, 0 <= m < b, \tag{3.2}$$

where m is the burn-in iteration, and b is 56, the number of burn-in iterations. Wang et al. (2013) do not describe this penalty variable, but it appears from inspection of the SNPTools source code that p_m bears resemblance to the inverse of an annealing temperature used in simulated annealing (SA) schemes described by Kirkpatrick, Gelatt, and Vecchi (1983). SA is commonly used with MH samplers to speed up burn-in of the MH sampler, so its use here is not surprising. For this reason we refer to the first 28 burn-in iterations as SA iterations.

In the Wang et al. (2013) implementation of the SNPTools model p_m has the following effects:

- It increases the MH acceptance probability defined in Equation 3.5. In the m^{th} burn-in iteration, the acceptance probability is

$$\min \left[1, \left(\frac{P(R|\Lambda')}{P(R|\Lambda)} \right)^{p_m} \right]. \tag{3.3}$$

- It flattens the distribution from which Y_l is sampled in case 2 (see subsubsection 3.1.4.2). For the first half of burn-in iterations, Y_l is not sampled with probability $g(v_i)$ defined in Equation 3.31, but instead with probability

$$g(v_i|p_m) = f(v_i)^{p_m} / \sum_{j=1}^{4} f(v_j)^{p_m}, \tag{3.4}$$

3.1.3.2 Surrogate parent sampling

This step is carried out M_2 times within each of M_1 iterations. As each individual is updated independently, it is sufficient to describe a single individual's update at the second level. Surrogate parent sampling consists of an MH sampler that aims to find a haplotype subset H_S that is consistent with R for an individual n. In the first iteration the model generates a proposal S at random and calculates $P(R|\Lambda = (H_S, r, \mu))$. The model then changes one of the elements of S to a random haplotype index in H, creating S', and calculates $P(R|\Lambda' = (H_{S'}, r, \mu))$. The new set S' is accepted with the MH acceptance probability of

$$\min\left[1, \frac{P(R|\Lambda')}{P(R|\Lambda)}\right]. \tag{3.5}$$

If S' is accepted, then S is replaced with S'. The next iteration consists of sampling a new S' based on S as described before, and performing another MH acceptance step as before. With default settings, SNPTools performs $M_2 = 2N$ MH steps for every individual for every one of M_1 iterations. The set S sampled last (at iteration M_2) is then used as in case 2 of subsection 3.1.4 to sample a new diplotype for individual n.

The MH sampling is used to sample from the posterior set of possible surrogate parents given the observed data R. If the surrogate parent haplotypes explain R well, then a new diplotype sampled from them with mutation will also be likely to explain R well.

3.1.4 The SNPTools HMM

For a single individual n, the SNPTools model uses an HMM as described by Rabiner (1989) to perform two functions:

Case 1 calculate $P(R|\Lambda)$, the probability of the observed reads given the HMM model parameters Λ

Case 2 sample L new pairs of alleles Y_l independently of each other from the marginal distribution $P(Y_l|R,\Lambda)$. This distribution is calculated with the help of the forward-backward algorithm (e.g., Durbin et al. (1998)).

3.1.4.1 Case 1

To calculate $P(R|\Lambda)$, Wang et al. (2013) assume that a sequence of markers can be modeled as a Markov chain with transition probabilities that depend on the per-generation genetic distances between loci $r = (r_1, \ldots, r_{L-1})$, hidden states that represent pairs of surrogate parent haplotypes from H_S, and emissions R.

The initial state of the Markov chain is uniform across all four possibilities

$$P(Z_1 = u_i) = 1/4, \quad 1 \leq i \leq 4. \tag{3.6}$$

Transition from SNP l to SNP $l+1$ is modeled as

$$P(Z_{l+1} = u_j | Z_l = u_i) = \begin{cases} (1-r_l)^2 & \text{if } i = j, \\ r_l^2 & \text{if } u_{i1} \neq u_{j1} \wedge u_{i2} \neq u_{j2}, \\ (1-r_l)r_l & \text{otherwise,} \end{cases} \tag{3.7}$$
$$1 \leq j \leq 4, \;\; 1 \leq i \leq 4,$$

where

$$r_l = \frac{d_l \rho}{2N + d_l \rho}. \tag{3.8}$$

Here, d_l is the genomic distance between site l and $l+1$, and the average recombination rate of a chunk is

$$\rho = 0.5 * \frac{L-1}{D \sum_{i=1}^{2N-1} 1/i}, \tag{3.9}$$

where D is the length in genomic coordinates between SNP 1 and SNP L. Wang et al. (2013) cite Fearnhead and Donnelly (2001) for their choice of r_l.

It is unclear from Wang et al. (2013) or their source code what motivated their choice of ρ (see Equation 3.9). However, ρ is related to Watterson's estimator of the population mutation rate $\hat{\theta}_w$ (Watterson, 1975) through the equation

$$\hat{\theta}_w = 2 * \frac{D}{L-1} L\rho. \tag{3.10}$$

Using an estimator of the population mutation rate to estimate recombination rates appears to be a mistake, but as we show later, the SNPTools model results do not appear to be very sensitive to the choice of ρ.

Wang et al. (2013) would like to quantify how well the observed data are supported by a sampled H_S in order to compare it to another sample of haplotypes, $H_{S'}$, in an MH step. This can be expressed as

$$P(R|\Lambda) = \sum_{\text{all } Z} P(R|Z,\Lambda)P(Z|\Lambda). \tag{3.11}$$

We wish to model an individual's diplotype as being similar, but not exactly the same as a mosaic of parental haplotypes referenced by Z. Therefore we introduce another latent variable Y representing the n^{th} individual's diplotype:

$$P(R|Z,\Lambda) = \sum_{allY} P(R|Y,Z,\Lambda)P(Y|Z,\Lambda). \tag{3.12}$$

Y is related to Z through mutation, and $P(Y|Z,\Lambda)$ is defined in Table 3.1, where the mutation rate

$$\mu = \frac{(\sum_{i=1}^{2N-1} 1/i)^{-1}}{2N + (\sum_{i=1}^{2N-1} 1/i)^{-1}}. \tag{3.13}$$

		Y_l			
		(0,0)	(0,1)	(1,0)	(1,1)
$H_{S_{u_i}l}$	(0,0)	$(1-\mu)^2$	$\mu(1-\mu)$	$\mu(1-\mu)$	μ^2
	(0,1)	$\mu(1-\mu)$	$(1-\mu)^2$	μ^2	$\mu(1-\mu)$
	(1,0)	$\mu(1-\mu)$	μ^2	$(1-\mu)^2$	$\mu(1-\mu)$
	(1,1)	μ^2	$\mu(1-\mu)$	$\mu(1-\mu)$	$(1-\mu)^2$

Table 3.1: *The probability of mutating between allele pairs.* This table describes the conditional probability $P(Y_l|Z_l = u_i)$. μ is defined in Equation 3.13.

In this model, reads are emitted from the individual's diplotype Y. The probability of emitting the observed reads from Y is quantified by the GLs. Therefore, R is conditionally independent of Z and Λ given Y, and

$$P(R|Y, Z, \Lambda) = P(R|Y). \tag{3.14}$$

Combining equations 3.11, 3.12, and 3.14 leads to

$$P(R|\Lambda) = \sum_{\text{all } Z} \sum_{\text{all } Y} P(R|Y)P(Y|Z, \Lambda)P(Z|\Lambda). \tag{3.15}$$

We assume that mutations arise at each site independently. This means that conditionally on Z, Y is independent of site:

$$P(Y|Z, \Lambda) = \prod_{l=1}^{L} P(Y_l|Z_l, \Lambda). \tag{3.16}$$

We also assume that observed reads are observed at each site independently:

$$P(R|Y) = \prod_{l=1}^{L} P(R_l|Y_l). \tag{3.17}$$

While this assumption simplifies the model, it should be noted that this assumption is certainly false because reads tend to span multiple sites.

Combining the previous three equations we get

$$P(R|\Lambda) = \sum_{\text{all } Z} \sum_{\text{all } Y} \left(\prod_{l=1}^{L} P(R_l|Y_l) P(Y_l|Z_l, \Lambda) \right) P(Z|\Lambda), \tag{3.18}$$

$P(R|\Lambda)$ in Equation 3.11 can be calculated efficiently using the forward backward algorithm of an HMM. Following Rabiner (1989), the forward variable $\alpha_l(i)$ is calculated[1] as follows:

1. Initialization:

$$\begin{aligned} \alpha_1(i) =& P(R_1|Z_1 = u_i, \Lambda) P(Z_1 = u_i|\Lambda) \\ =& P(R_1|Z_1 = u_i, \Lambda)\ 1/4, \quad 1 \leq i \leq 4 \end{aligned} \tag{3.19}$$

2. Induction:

$$\alpha_{l+1}(j) = \left(\sum_{i=1}^{4} \alpha_l(i) P(Z_{l+1} = u_j|Z_l = u_i) \right) P(R_{l+1}|Z_{l+1} = u_j, \Lambda), \tag{3.20}$$
$$1 \leq j \leq 4, \quad 1 \leq l < L.$$

3. Termination:

$$P(R|\Lambda) = \sum_{i=1}^{4} \alpha_L(i). \tag{3.21}$$

3.1.4.2 Case 2

Once a suitable set H_S has been sampled, a new diplotype Y is sampled from the HMM. Wang et al. (2013) use the forward-backward algorithm (see for example Rabiner (1989)) to calculate and sample from the marginal state probabilities $P(Y_l|R, \Lambda)$. Each Y_l is sampled independently. In contrast, other HMM-based phasing models

[1] In their implementation, Wang et al. (2013) actually calculate the forward probabilities from the back and the backward probabilities from the front, but that is only an implementation choice and does not affect the mathematics of the model described here.

(Delaneau, Zagury, and Marchini, 2013; Y. Li, Willer, et al., 2010) employ a forward-backtrack sampling scheme to sample new haplotypes. See subsubsection 3.1.5.3 for a formulation of case 2 that uses the forward-backtrack sampling scheme according to Cawley and Pachter (2003).

After introducing the same latent variable Z as in case 1,

$$\begin{aligned} P(Y_l|R,\Lambda) =& \sum_{allZ} P(Y_l, Z|R,\Lambda) \\ =& \sum_{allZ} P(Y_l|Z,R,\Lambda)P(Z|R,\Lambda). \end{aligned} \tag{3.22}$$

Because we assume that reads are emitted from Y_l at each site independently, and that mutations arise at each site independently (see Equation 3.16 and Equation 3.17), Y_l is independent of all other R and Z conditional on Z_l and R_l:

$$\begin{aligned} P(Y_l|R,\Lambda) =& \sum_{allZ} P(Y_l|Z_l,R_l,\Lambda)P(Z|R,\Lambda) \\ =& \sum_{i=1}^{4} \sum_{allZ,where Z_l=u_i} P(Y_l|Z_l=u_i,R_l,\Lambda)P(Z|R,\Lambda) \\ =& \sum_{i=1}^{4} P(Y_l|Z_l=u_i,R_l,\Lambda) \sum_{allZ,where Z_l=u_i} P(Z|R,\Lambda) \\ =& \sum_{i=1}^{4} P(Y_l|Z_l=u_i,R_l,\Lambda)P(Z_l=u_i|R,\Lambda) \\ =& \sum_{i=1}^{4} P(Y_l|Z_l=u_i,R_l,\Lambda)P(Z_l=u_i,R|\Lambda)/P(R|\Lambda), \end{aligned} \tag{3.23}$$

where

$$\begin{aligned} P(Y_l|Z_l=u_i,R_l,\Lambda) =& P(Y_l,R_l|Z_l=u_i,\Lambda)/P(R_l|Z_l=u_i,\Lambda) \\ =& P(R_l|Y_l)P(Y_l|Z_l=u_i,\Lambda)/P(R_l|Z_l=u_i,\Lambda). \end{aligned} \tag{3.24}$$

The result of combining equations Equation 3.23 and Equation 3.24 is

$$\begin{aligned} P(Y_l|R,\Lambda) =& \sum_{i=1}^{4} \frac{P(R_l|Y_l)P(Y_l|Z_l=u_i,\Lambda)P(Z_l=u_i,R|\Lambda)}{P(R|\Lambda)P(R_l|Z_l=u_i,\Lambda)} \\ =& \frac{P(R_l|Y_l)}{P(R|\Lambda)} \sum_{i=1}^{4} \frac{P(Y_l|Z_l=u_i,\Lambda)P(Z_l=u_i,R|\Lambda)}{P(R_l|Z_l=u_i,\Lambda)} \end{aligned} \tag{3.25}$$

$P(R_l|Y_l)$ is defined in Equation 3.1. $P(Y_l|Z_l,\Lambda)$ is given in Table 3.1, and $P(R|\Lambda)$ can be calculated using Equation 3.21. To calculate $P(Z_l=u_i,R|\Lambda)$ we require both the forward probabilities $\alpha_l(i)$ and the backward probabilities

$$\beta_l(i) = P(R_{l+1},\ldots,R_L|Z_l=u_i,\Lambda), \tag{3.26}$$

which is the probability of the observed reads from site $l+1$ to L given state Z_l and the model Λ. Like the forward probabilities, the backward probabilities are calculated inductively:

1. Initialization:

$$\beta_L(i) = 1, \quad 1 \le i \le 4 \tag{3.27}$$

2. Induction:

$$\begin{aligned} \beta_l(i) =& \sum_{j=1}^{4} P(R_{l+1}|Z_{l+1}=u_j)\beta_{l+1}(j)P(Z_l=u_i|Z_{l+1}=u_j) \\ & 1 \le j \le 4,\ l = L-1, L-2, \ldots, 1. \end{aligned} \tag{3.28}$$

It should be noted that because the transition matrix is symmetric, $P(Z_l=u_j|Z_{l+1}=u_i) = P(Z_{l+1}=u_i|Z_l=u_j)$.

Once $\beta_l(i)$ has been calculated, it is easy to calculate the joint probability of R

and $Z_l = u_i$:

$$P(Z_l = u_i, R|\Lambda) = \alpha_l(i)\beta_l(i). \tag{3.29}$$

And Equation 3.25 becomes

$$P(Y_l|R,\Lambda) = \frac{P(R_l|Y_l)}{P(R|\Lambda)} \sum_{i=1}^{4} \frac{P(Y_l|Z_l = u_i, \Lambda)\alpha_l(i)\beta_l(i)}{P(R_l|Z_l = u_i, \Lambda)}. \tag{3.30}$$

We sample $Y_1, \ldots, Y_L$ with probability proportional to Equation 3.30, and replace the individual's diplotype with Y.

3.1.5 Critique of the SNPTools HMM

There are two points to make about the sampling of Y as described in subsubsection 3.1.4.2. Both of these aspects of the SNPTools are not made clear in Wang et al. (2013), but instead were discovered while analyzing the source code of SNPTools.

3.1.5.1 The SNPTools implementation deviates from the model

The SNPTools implementation by Wang et al. (2013) deviates from the model described in subsubsection 3.1.4.2. Instead of sampling from $P(Y_l|R,\Lambda)$ as described in Equation 3.30, the SNPTools implementation samples a v_i for Y_l with probability

$$g(v_i) = f(v_i) / \sum_{j=1}^{4} f(v_j), \tag{3.31}$$

where

$$f(v_i) = P(R_l|Y_l = v_i) \sum_{j=1}^{4} P(Y_l|Z_l = u_j)\hat{\alpha}_l(j)\delta_l(j). \tag{3.32}$$

Here, $\alpha_l(j)$ is scaled for numerical stability as described in Rabiner (1989):

$$\hat{\alpha}_l(i) = \left(\prod_{\tau=1}^{l} c_\tau \right) \alpha_l(i), \tag{3.33}$$

where

$$c_l = \frac{1}{\sum_{j=1}^{4} \alpha_l(j)}. \tag{3.34}$$

δ is defined by recursion:

1. Initialization:

$$\delta_L(i) = 1, \quad 1 \leq i \leq 4 \tag{3.35}$$

2. Induction:

$$\delta'_l(j) = \sum_{i=1}^{4} \delta_{l+1}(i) P(R_{l+1}|Z_{l+1} = u_i) P(Z_l = u_j|Z_{l+1} = u_i) \tag{3.36}$$

$$\delta_l(j) = \delta'_l(j) / \sum_{i=1}^{4} \delta'_l(i) \tag{3.37}$$

$$1 \leq j \leq 4, \; l = L-1, L-2, \ldots, 1.$$

The induction step for $\delta_l(i)$ differs from the induction step for $\beta_l(i)$ only in the extra scaling step of Equation 3.37. This is an incorrect scaling. According to Rabiner (1989), the correct scaling of $\beta_l(i)$ is

$$\hat{\beta}_l(i) = c_l \beta_l(i), \tag{3.38}$$

and in Equation 3.32 $\delta_l(i)$ should be replaced by $\hat{\beta}_l(i)$:

$$f(v_i) = P(R_l|Y_l = v_i) \sum_{j=1}^{4} P(Y_l|Z_l = u_j) \hat{\alpha}_l(j) \hat{\beta}_l(j). \tag{3.39}$$

However, the correctness of the scaling factor does not matter in this sampling scheme because either scaling factor is canceled out in Equation 3.31.

The other difference between Equation 3.31 and Equation 3.30 is that the first equation lacks the divisor $P(R_l|Z_l = u_i,)$ inside the sum.

Note also the deviation described in subsubsection 3.1.3.1: During the SA burn-in iterations, the Wang et al. (2013) SNPTools implementation samples from Equation 3.4 instead of from Equation 3.31.

3.1.5.2 The sampling scheme for Y can lead to inconsistent haplotypes

In the case 2 the SNPTools model samples all Z_l independently of each other. In theory, this can lead to the sampling of a state sequence Z that is inconsistent with the model Λ.

A pathological example might look like this: Let us assume four loci ($L = 4$) and that the parental haplotypes H_S and observed GLs $P(R_l|G_l)$ are as described in Table 3.2. A path Z consistent with H_S and the GLs would be $Z^1 = ((2,3),(2,3),(2,3),(2,3))$, which is the same as copying only from H_{S_2} and H_{S_3}. However, when using Equation 3.30 as a sampling scheme, another path that is equally likely is $Z^2 = ((2,3),(1,4),(2,3),(1,4))$. The resulting sampled haplotypes are described in Table 3.2. However, the path Z^1 is more consistent with the parental haplotypes because it requires no recombination events, whereas Z^2 requires three recombination events.

However, this site-independent sampling scheme is less of a problem if the surrogate parents chosen by the surrogate parent sampling scheme (subsubsection 3.1.3.2) are mostly homozygous. A second example shows that this might be the case. Let us compare $P(R|H_S, r, \mu)$ and $P(R|H_{S'}, r, \mu)$, where H_S and the GLs are given in Table 3.2, and $H_{S'}$ is described in Table 3.3. S' represents a reordering of S. Under S' every possible path through the HMM is consistent with the GLs, whereas under S there

	l			
	1	2	3	4
H_{S_1}	1	1	1	1
H_{S_2}	0	0	0	0
H_{S_3}	1	1	1	1
H_{S_4}	0	0	0	0

	l			
	1	2	3	4
$P(R_l\|G_l=0)$	0	0	0	0
$P(R_l\|G_l=1)$	1	1	1	1
$P(R_l\|G_l=2)$	0	0	0	0

	l			
	1	2	3	4
Z^1	0	0	0	0
	1	1	1	1
Z^2	1	0	1	0
	0	1	0	1

Table 3.2: *Pathological example.* **Left:** four haplotypes making up H_S. **Center:** GLs for sites $1, \ldots, L$. **Right:** Diplotypes of two equally likely sampled state sequences Z^1 and Z^2 under the sampling scheme described in Equation 3.30.

	l			
	1	2	3	4
$H_{S'_1}$	1	1	1	1
$H_{S'_2}$	1	1	1	1
$H_{S'_3}$	0	0	0	0
$H_{S'_4}$	0	0	0	0

Table 3.3: *Four haplotypes making up $H_{S'}$.*

are several paths that are inconsistent with the GLs. $P(R|H_S, r, \mu) < P(R|H_{S'}, r, \mu)$, and the surrogate parent sampling scheme would be more likely to sample $H_{S'}$ than H_S. This may explain why the phasing performance of the SNPTools model is not substantially worse.

3.1.5.3 Forward-backtrack sampling

When sampling a path through an HMM, instead of sampling each state independently as SNPTools does, other phasing models (Delaneau, Zagury, and Marchini, 2013; Y. Li, Willer, et al., 2010) employ what Cawley and Pachter (2003) call a forward-backtrack algorithm. In the SNPTools case we sample a new diplotype Y from the distribution $P(Y|R, \Lambda)$.

After introducing the same latent variable Z as in case 1,

$$\begin{aligned} P(Y|R,\Lambda) =& \sum_{allZ} P(Y,Z|R,\Lambda) \\ =& \sum_{allZ} P(Y|Z,R,\Lambda)P(Z|R,\Lambda), \end{aligned} \tag{3.40}$$

we may sample Z using the forward-backtrack algorithm described by Cawley and Pachter (2003). Exploiting the Markov property of the model, we rewrite $P(Z|R,\Lambda)$ as

$$\begin{aligned} P(Z|R,\Lambda) =& P(Z_L|R,\Lambda)\prod_{l=1}^{L-1} P(Z_l|Z_{l+1},R,\Lambda) \\ =& P(Z_L|R,\Lambda)\prod_{l=1}^{L-1} P(Z_l|Z_{l+1},R_1,\ldots,R_l,\Lambda). \end{aligned} \tag{3.41}$$

We may now sample Z one state at a time. Assuming the forward variables $\alpha_l(i)$ have already been calculated as in case 1, we sample Z in the following manner:

1. Initialization:

   Sample a u_i for Z_L with probability

$$\begin{aligned} P(Z_L = u_i|R,\Lambda) =& P(Z_L = u_i, R|\Lambda)/P(R|\Lambda) \\ =& \alpha_L(i)/\sum_{i=1}^{4}\alpha_L(i). \end{aligned} \tag{3.42}$$

2. Induction:

Sample a u_j for Z_l conditional on $Z_{l+1} = u_i$ with probability

$$\begin{aligned} P(Z_l|Z_{l+1}, R_1, \ldots, R_l, \Lambda) \propto & P(Z_l, Z_{l+1}, R_1, \ldots, R_l|\Lambda) \\ = & P(Z_{l+1}|Z_l, \Lambda)\ P(Z_l|R_1, \ldots, R_l, \Lambda) \\ = & \frac{P(Z_{l+1}|Z_l, \Lambda)\ P(Z_l, R_1, \ldots, R_l|\Lambda)}{P(R_1, \ldots, R_l|\Lambda)} \\ = & \frac{P(Z_l|Z_{l+1}, \Lambda)\ P(Z_l, R_1, \ldots, R_l|\Lambda)}{P(R_1, \ldots, R_l|\Lambda)} \\ = & \frac{P(Z_l|Z_{l+1}, \Lambda)\ \alpha_l(j)}{\sum_{j=1}^{4} \alpha_l(j)}, \\ & 1 \leq l < L. \end{aligned} \tag{3.43}$$

It should be noted that because the transition matrix is symmetric, $P(Z_l = u_j|Z_{l+1} = u_i) = P(Z_{l+1} = u_i|Z_l = u_j)$.

After sampling Z we may sample Y conditional on Z. As described in case 1, conditional on Z, Y_l is independent of reads and states at other sites:

$$P(Y|Z, R, \Lambda) = \prod_{l=1}^{L} P(Y_l|R_l, Z_l, \Lambda), \tag{3.44}$$

where $P(Y_l|R_l, Z_l, \Lambda)$ is defined in Equation 3.24. To sample Y we sample Y_l at each site with probability proportional to $P(Y_l|R_l, Z_l, \Lambda)$.

3.2 Addition of a reference panel

My work with the SNPTools model has focused on increasing its speed because even though the SNPTools model can be more accurate than BEAGLE (see Figure 3.1), SNPTools is much too slow to be practically applied to large sample sizes (see Figure 3.3). From Pasaniuc et al. (2012) and my work on the CONVERGE project (see chapter 2) we know that using a reference panel substantially improves genotype

imputation accuracy when using BEAGLE. To explore if adding a reference panel to SNPTools is also beneficial, we extend the Wang et al. (2013) implementation of the SNPTools model to allow genotype calling and phasing with the help of a reference panel.

After initializing H, we augment this haplotype set by M reference haplotypes, and allow the surrogate parent haplotype indices of S to be sampled from $\{1, \ldots, 2N, \ldots, 2N+M\}$. To reduce burn-in iterations we can further restrict S to only be sampled from $\{2N+1, \ldots, 2N+M\}$ in the first iteration. In theory this should eliminate the need to use as many burn-in M_1 iterations as SNPTools, which uses 23 SA and 23 non-SA M_1 iterations. Results presented in subsection 3.2.1 use only 3 non-SA burn-in iterations, and the results are competitive with BEAGLE.

In the original SNPTools code the number of M_2 iterations cannot be reduced below $2N$. In order to explore reducing the number of M_2 iterations, the SNPTools code was also altered to allow the specification of the number of M_2 iterations to perform per individual. We refer to the SNPTools code containing all my alterations as "GLPhase".

3.2.1 Genotyping accuracy experiments

As part of the experiments to decide on how to call genotypes on the CONVERGE project (see chapter 2), we investigated the genotype calling accuracy of SNPTools v1.0, BEAGLE v3.3.2, and GLPhase using a reference panel.

The data set consisted of 9,300 CONVERGE samples for which we had called genotype likelihoods using SNPTools at all sites polymorphic in 1000 Genomes Phase 1 (The 1000 Genomes Project Consortium, 2012) Asian samples (Chinese and Japanese). We randomly selected five non-overlapping regions on chromosome 20 consisting of 1024 bi-allelic SNPs[2], and phased the GLs for the CONVERGE samples using the

[2]1024 sites is the default chunk size for SNPTools

following methods:

- BEAGLE with default arguments (10 iterations)
- SNPTools with default arguments (256 M_1 iterations and 18,600 M_2 iterations per individual)
- GLPhase with a reference panel using 25 M_1 iterations (3 non-SA burn-in + 22 sampling) and 2,000 M_2 iterations
- GLPhase with a reference panel using 10 M_1 iterations (3 burn-in and 7 sampling) and 100 M_2 iterations

Each of these methods were run without a reference panel, and the methods that allowed a reference panel were run with 1000 Genomes Phase 1 Asian reference panel (566 haplotypes) or with the complete 1000 Genomes Phase 1 reference panel (2178 haplotypes). The run time and RAM usage of each method were noted. Genotype calling accuracy of each method was also assessed by calculating the aggregate Pearson's R^2 between genotype calls made by each method to genotype calls from an Illumina Zhonghua-8 900k SNP chip we had available for 16 of the CONVERGE samples. The results are presented in Figure 3.1.

From the results of these experiments we drew the following conclusions:

- SNPTools with default arguments calls genotypes better than BEAGLE without a reference panel, but is considerably slower.
- BEAGLE attains similar genotyping accuracy to SNPTools with default arguments when using a reference panel.
- There appears to be no benefit to any of the investigated methods of using the full 1000 Genomes Phase 1 reference panel over the Asian subset of that panel.

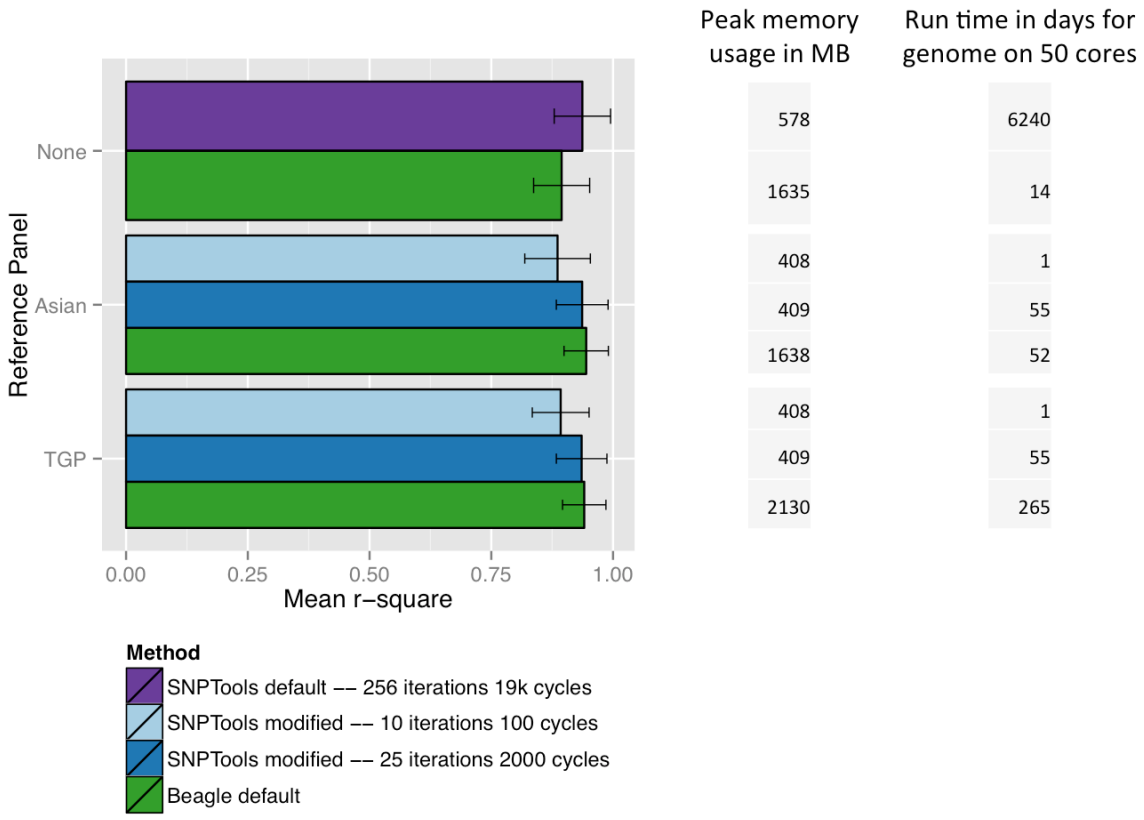

Figure 3.1: *Mean R^2, peak memory usage, and run time of SNPTools, BEAGLE 3.3, and GLPhase using a reference panel.* R^2 was calculated between imputed genotypes and Zhonghua-8 SNP chip genotypes in 16 CONVERGE samples. Error bars indicate two standard deviations below and above the mean. "iterations" refers to M_1 iterations, and "cycles" refers to M_2 iterations. "SNPTools modified" refers to GLPhase.

- GLPhase with 25 M_1 iterations and an Asian reference panel comes close to the genotyping accuracy of unmodified SNPTools with default arguments at two orders of magnitude less run time.

- BEAGLE with a reference panel is the top performing method.

Based on this data we chose to run BEAGLE with an Asian 1000 Genomes Phase 1 reference panel to call genotypes for CONVERGE. However, it was encouraging that modification of the SNPTools model in this way could lead to large speed gains at little cost in accuracy.

3.3 Using pre-existing haplotypes

The HRC data set is similar to the CONVERGE data set in that it consists of genotype likelihoods of low coverage sequencing data. However, unlike CONVERGE, the HRC consists of 20 groups that have already called their own haplotypes from their data. These pre-existing haplotypes posed an opportunity to save on compute time if the model used to call genotypes could be adapted to take advantage of these haplotypes. It was evident from the results of my testing of BEAGLE (see subsection 3.2.1 and Figure 3.1) that RAM usage and run time of BEAGLE increase with reference panel size. Based on the run time of BEAGLE on a 1024 site chunk of chromosome 20 of the HRC pilot data (12,753 samples), we estimated that BEAGLE using no reference panel would need 94 CPU years to complete the calling of about 50 million sites. We estimated that we would only have about 40 CPU years of compute to phase more than 30,000 samples. Furthermore, BEAGLE also scales super-linearly with sample size (B. L. Browning and S. R. Browning, 2007). Based on these pieces of information, it was clear that we needed to come up with a faster method that preferably scales better with sample size. The results of running GLPhase presented in Figure 3.1 were encouraging because they show that by altering SNPTools it is

possible to reduce run time substantially while obtaining genotype imputation accuracy comparable to BEAGLE, and we decided we should try and develop a method based on SNPTools for calling genotypes in the HRC.

For every individual to be phased and imputed, SNPTools uses an MH sampler to generate a set of four surrogate parent haplotypes with which to phase the individual. My extension uses pre-existing haplotypes to restrict the set of possible haplotypes from which the MH sampler may sample surrogate parent haplotypes. Specifically, for every individual to be phased and imputed, within a chunk of 1024 sites (this is the default), my algorithm performs a k-nearest neighbor (k-NN) search in which every pre-existing haplotype is compared to every other pre-existing haplotype using Hamming distance at the sites falling within the 1024 site chunk being phased. A list of samples to restrict surrogate parents is constructed in the following way:

1. The k-NN distance matrix (excluding the pre-existing haplotypes of the sample to be phased) is used to create a list containing the closest $k/2$ pre-existing haplotypes to the first of the two pre-existing haplotypes of the sample being phased.

2. The k-NN distance matrix (excluding the pre-existing haplotypes of the sample to be phased) is used to create a second list containing all pre-existing haplotypes ordered by distance between them and the second pre-existing haplotype of the sample to be phased.

3. Both lists are converted to a list of sample IDs of the sample that each haplotype belongs to.

4. Starting from the beginning of the second list, sample IDs that do not occur in the first list are appended to the first list until the first list contains k sample IDs. This list is the final list of sample IDs that may be used as surrogate parents

for the sample being phased[3], and it is created once before any iterations are run.

After restricting the set of possible surrogate parents in this way, my method performs efficiently using five non-SA burn-in M_1 iterations, 95 sampling M_1 iterations, and 100 M_2 iterations, and it outperforms the original SNPTools algorithm when used with these parameter choices (Compare "CPU" and "No pre-existing haps" in Figure 3.2). The SNPTools parameter defaults are 56 burn-in M_1 iterations, 200 sampling M_1 iterations, and $2 * N$ M_2 iterations. An exploration of my parameter choices is given in section 3.5. These parameter choices reduce the complexity of my phasing algorithm from $O(N^2)$ to $O(N)$.

Although my implementation of the nearest neighbor search is complexity $O(N^2)$, for N = 30,000 the impact of the k-NN search on run time is small. Only about ten out of about 70 minutes of run time are spent on the k-NN search. Once sample sizes are encountered where the k-NN search begins to dominate, my implementation could be replaced with another algorithm that has better complexity. Examples of such algorithms, both with complexity $O(N \log N)$ are one by Vaidya (1989), and one based on k-means clustering developed by O'Connell, Sharp, et al. (2016).

To demonstrate the scaling of SNPTools, BEAGLE v3.3.2, BEAGLE v4.1, and GLPhase v1.4.13, we randomly selected five 1024 site regions on chromosome 20 from the final HRC site list. The regions are given in Table B.1. We subset the final HRC GLs to decreasing random subsets of samples down to 500 samples, and ran each of the four programs on the GLs. The runs were performed in single-threaded mode on 16 core Intel Xeon CPU E5-2650 v2 @ 2.60GHz cluster nodes running in non-hyperthreaded mode. Both BEAGLE programs were run using Java v1.8.0_45.

[3]It should be noted that in the case of overlaps between the first $k/2$ sample IDs in the two lists, this scheme gives more weight to the second pre-existing haplotype than the first. A symmetric algorithm would randomly choose from the first or the second list when appending to the final list in the case of a sample ID overlap.

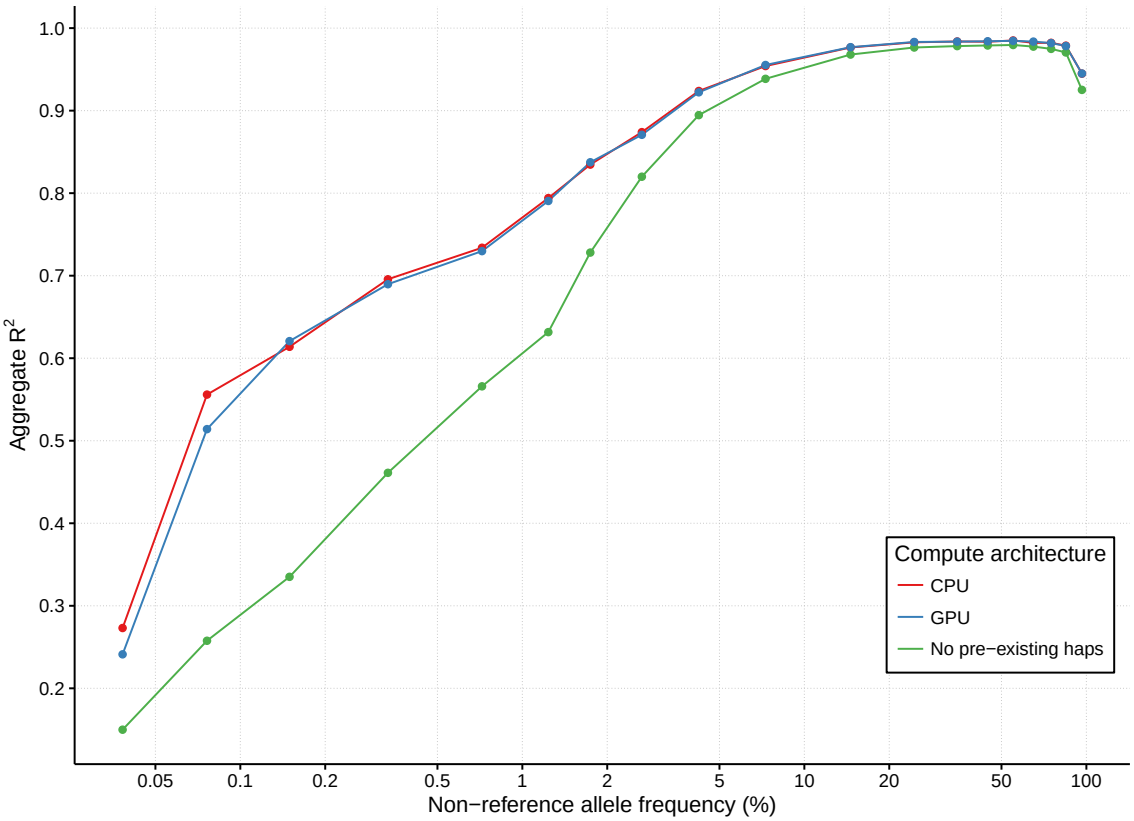

Figure 3.2: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase, the CUDA implementation of GLPhase, and SNPTools, using GLPhase default parameters.* These haplotypes were not rephased with SHAPEIT. The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs. "No pre-existing haps" is the original SNPTools model run with the (reduced) iterations used for GLPhase. Genotype discordances of all of these panels with an OMNI 2.5M chip are listed in Table 3.4.

The run time estimates are displayed in Figure 3.3. It appears that SNPTools and both BEAGLE versions empirically scale $O(N^2)$ for the range of sample subsets examined. GLPhase appears to scale linearly until about 10,000 samples, at which point the quadratic part of the algorithm starts to have a small, but noticeable impact.

3.4 Genetic map

Wang et al. (2013) choice for ρ is problematic because it has no theoretical basis. We introduce an option to use a genetic map for calculating ρ. Initially, this was implemented erroneously in such a way that the genetic map distances were inflated by a factor of 100. The error was discovered during code review and fixed by applying a "genetic map fix". We found that the fix changed genotyping accuracy (see Table 3.4). Heterozygous sites are genotype better, whereas homozygous genotypes are genotyped worse. However, downstream imputation accuracy from these haplotypes is similar to the downstream imputation accuracy of the haplotypes created by the original GLPhase implementation (see Figure 3.4). After rephasing with SHAPEIT3 (see Figure 3.5), these haplotypes have a genotype imputation accuracy that is similar to the imputation accuracy of haplotypes generated from the original GLPhase model that have been rephased with SHAPEIT3. We conclude that the choice of ρ in GLPhase matters somewhat for genotyping accuracy. However, the change in genotyping accuracy does not appear to propagate to downstream imputation accuracy from the haplotypes produced by GLPhase.

3.5 Investigation of GLPhase parameter choices

With the introduction of clustering on pre-existing haplotypes, choices of M_1, M_2, and k for the k-NN search had to be made. Figure 3.6 and Figure 3.7 show that changing the number of M_1 non-SA burn-in or sampling iterations, respectively, does not appear

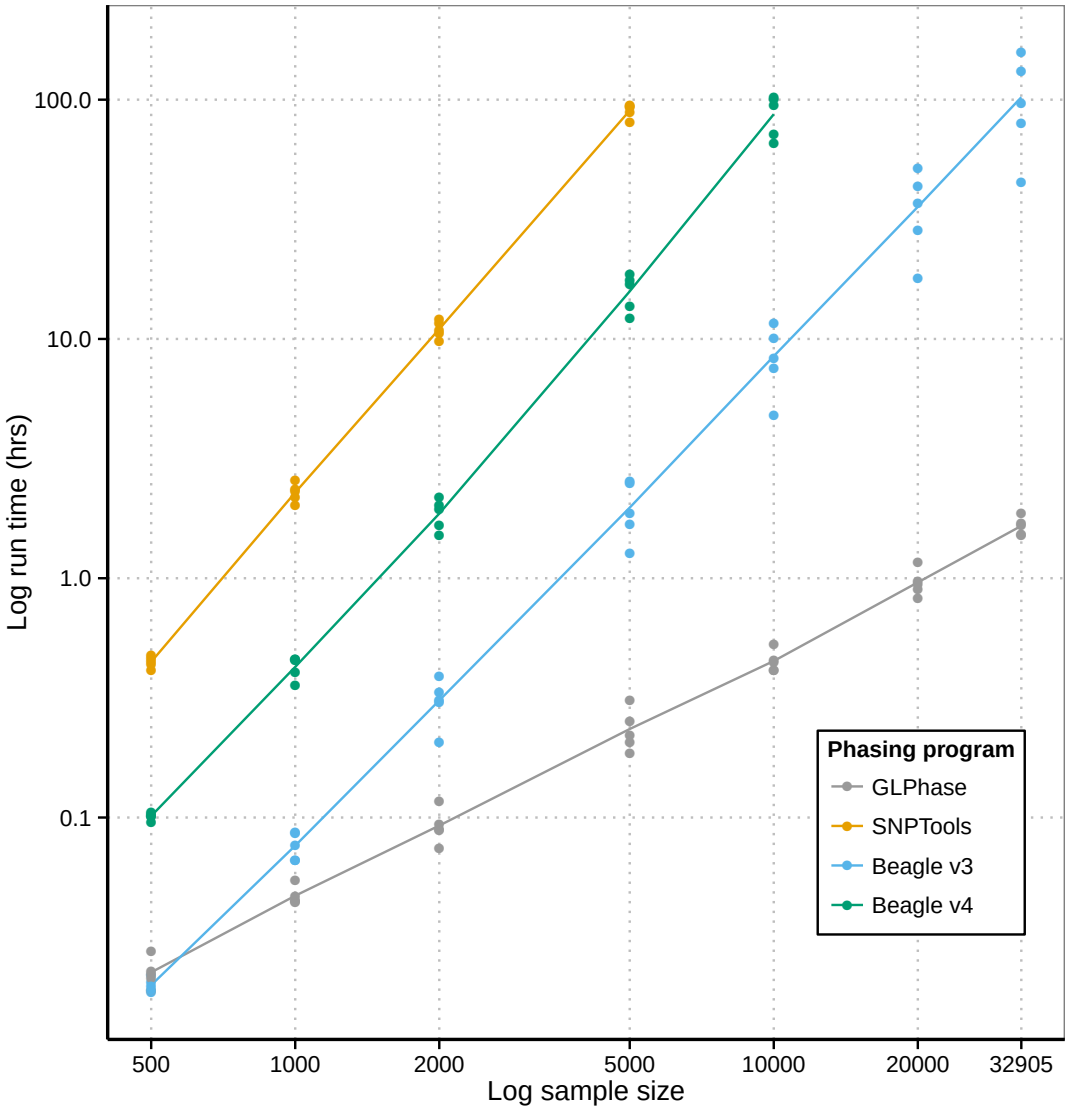

Figure 3.3: *Log-log plot of genotype calling program run times versus sample size.* Run times were calculated by calling genotypes from GLs in five chunks of 1024 sites from the final HRC GL set (see Table B.1). GLs were subset to random subsets of the HRC samples.

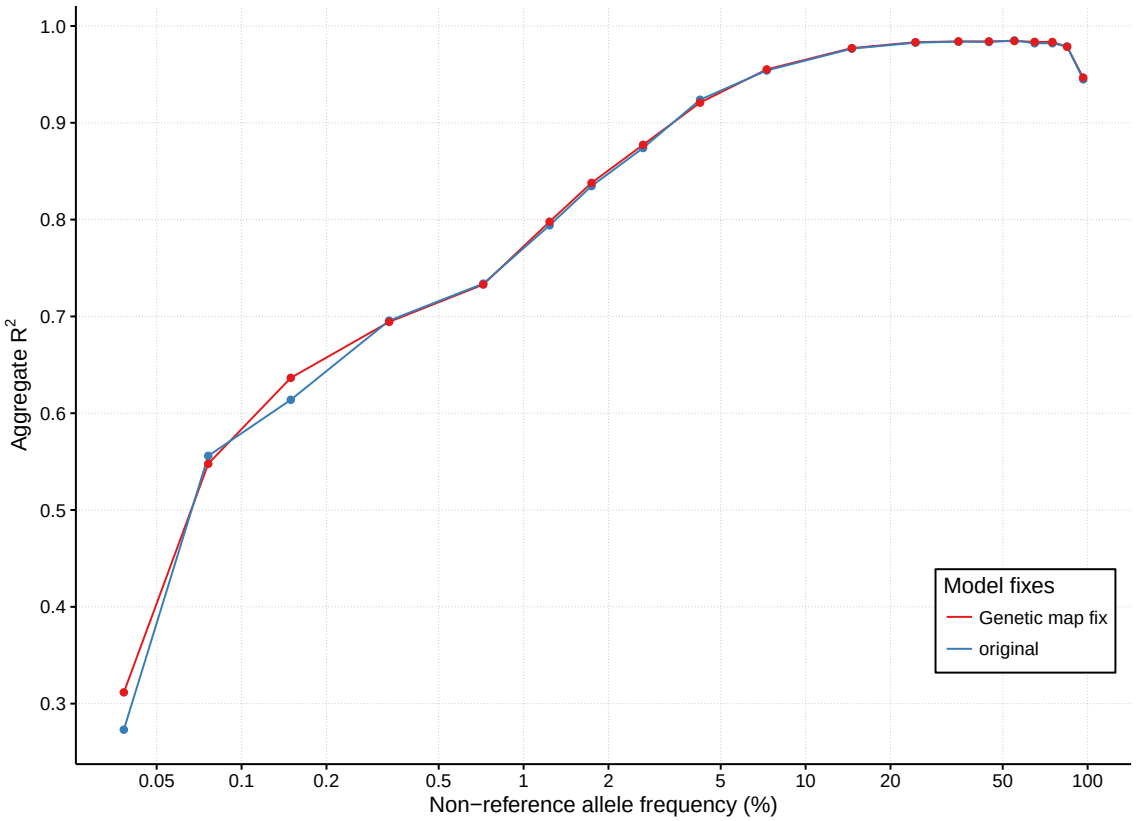

Figure 3.4: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase after implementing the genetic map fix described in section 3.4.* These haplotypes were not rephased with SHAPEIT. The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs on chromosome 20. "original" refers to GLPhase v1.4.13. The genetic map fix improves downstream imputation accuracy by a small amount. Genotype discordances of these panels with an OMNI 2.5M chip are listed in Table 3.5.

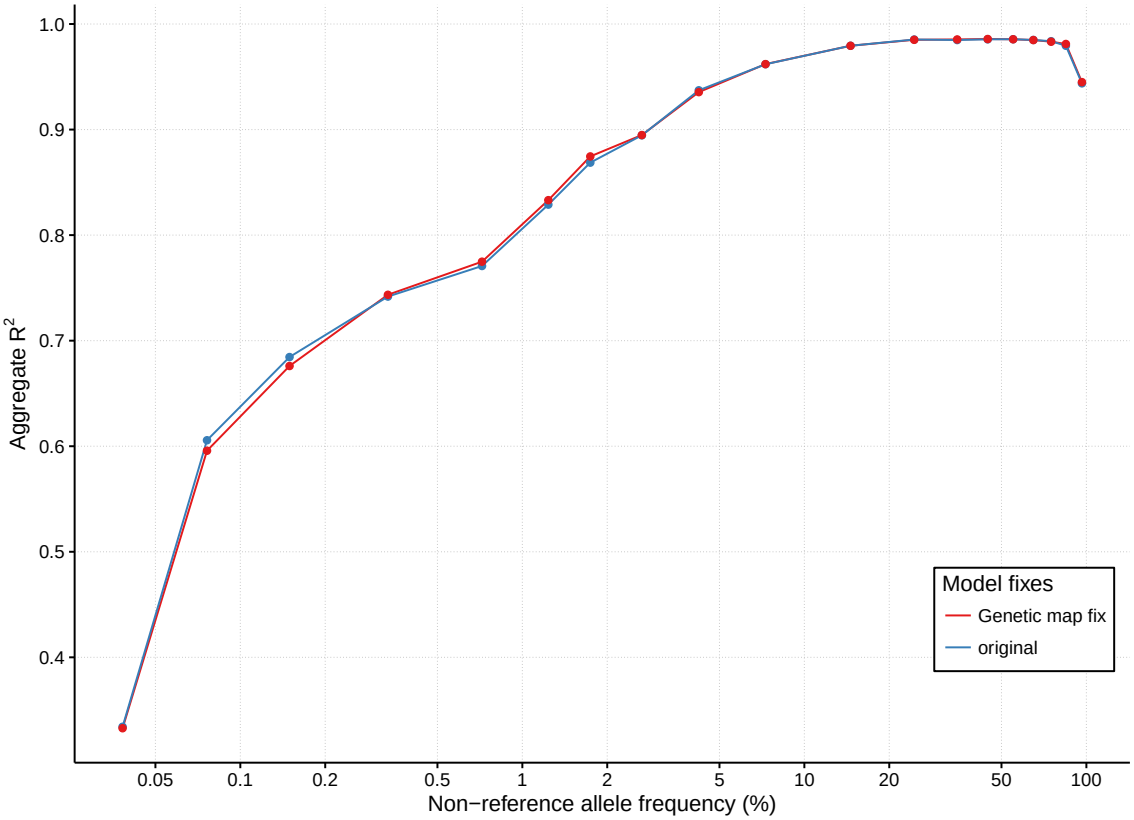

Figure 3.5: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase after implementing the genetic map fix described in section 3.4 and rephasing with SHAPEIT3.* The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs on chromosome 20. "original" refers to GLPhase v1.4.13. After rephasing the genetic map fix does not appear to change downstream imputation accuracy.

Changes from default GLPhase	NRD	RRD	RAD	AAD
none	0.3843	0.0549	0.3365	0.2321
Genetic map fix	0.4284	0.0887	0.2741	0.3228
GPU implementation	0.4621	0.0874	0.3071	0.3634
No pre-existing haplotypes	1.2346	0.1773	1.0787	0.7523

Table 3.4: *Genotype discordance (in percent) between OMNI 2.5M genotypes available for 364 European samples from the 1000 Genomes project and haplotype panels shown in Figure 3.4 and Figure 3.2 (GPU implementation, and No pre-existing haplotypes).* Discordances were measured at 42,229 SNPs on chromosome 20. The columns labeled "NRD", "RRD", "RAD", and "AAD" respectively contain the non-reference, homozygous reference allele, heterozygous, and homozygous alternate allele genotype discordances. "No pre-existing haplotypes" is the original SNPTools model run with the (reduced) iterations used for "none". Applying the genetic map fix lowers the RAD rate, but increases all other discordance rates.

to change the quality of haplotypes as measured by downstream imputation accuracy. For M_2 iterations the story is similar, except that reduction to 50 iterations appears to reduce haplotype quality as described in Figure 3.8. It should be noted that an increase of M_2 iterations beyond about 400 iterations would have made running GLPhase on the HRC data infeasible due to our run time constraints. Within the range of 25 to 400 the choice of the number of nearest neighbors (k) also does not appear to matter for downstream imputation accuracy (see Figure 3.9). However decreasing k to 16 does reduce downstream imputation accuracy.

From the discordance rates listed in Table 3.5 it appears that decreasing M_2 iterations also decreases genotyping accuracy, whereas decreasing M_1 sampling and non-SA burn-in iterations appears not to affect genotyping accuracy by as much. Both M_1 and M_2 iterations appear to increase genotyping accuracy. However, doubling both M_1 and M_2 iterations has an equal computational cost, so better genotypes may be obtainable for the same computational cost by decreasing M_1 and increasing M_2 iterations.

Decreasing the number of nearest neighbors used by GLPhase in clustering appears to increase genotyping accuracy (Table 3.5). However, it should be noted that the

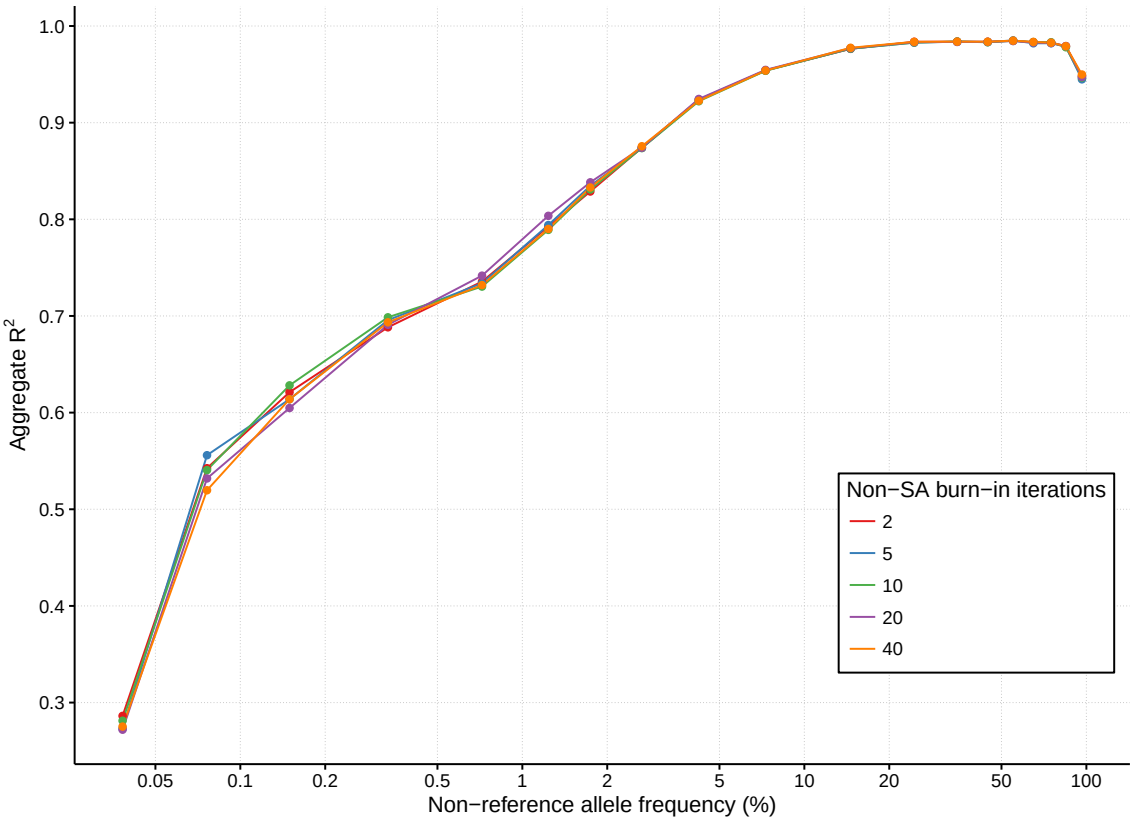

Figure 3.6: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase while varying the number of M_1 non-SA burn-in iterations on chromosome 20.* These haplotypes were not rephased with SHAPEIT. The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs. Non-SA M_1 burn-in iterations between 2 and 40 appear to result in haplotypes with similar downstream imputation accuracy. Genotype discordances of all of these panels with an OMNI 2.5M chip are listed in Table 3.5.

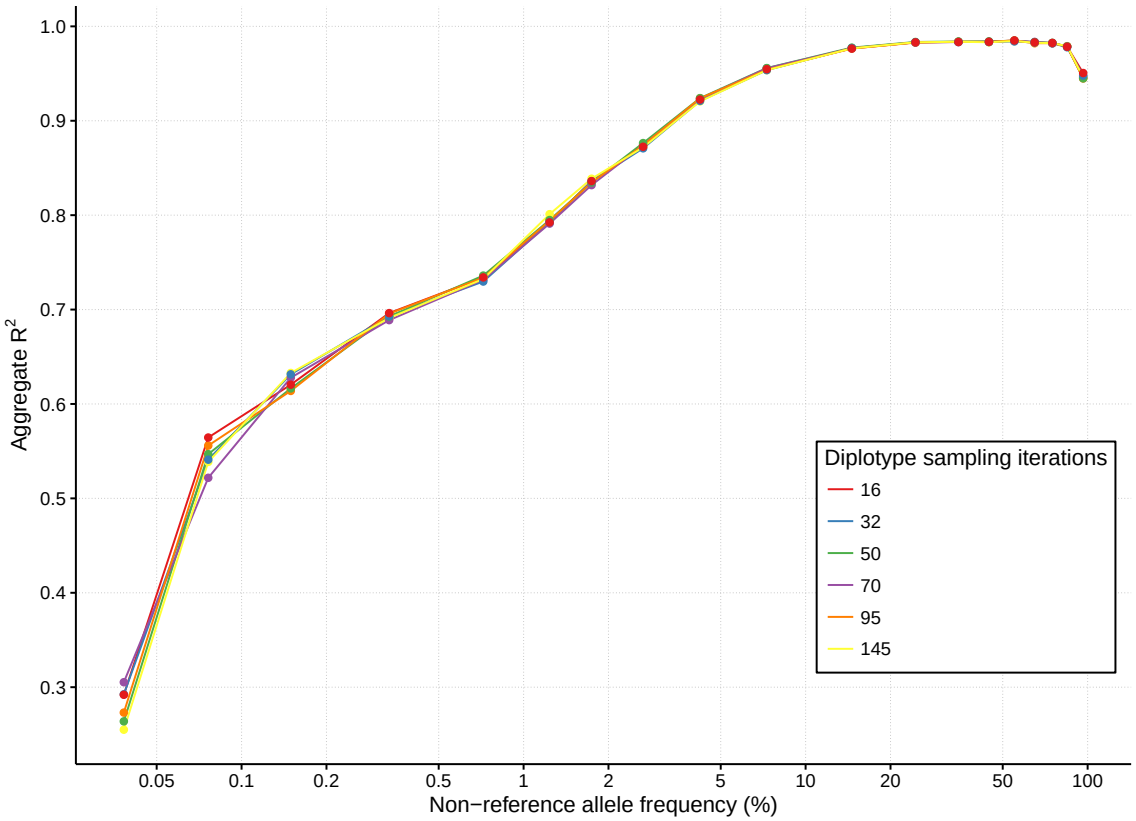

Figure 3.7: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase while varying the number of M_1 sampling iterations on chromosome 20.* These haplotypes were not rephased with SHAPEIT. The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs. M_1 sampling iterations between 16 and 145 appear to result in haplotypes with similar downstream imputation accuracy. Genotype discordances of all of these panels with an OMNI 2.5M chip are listed in Table 3.5.

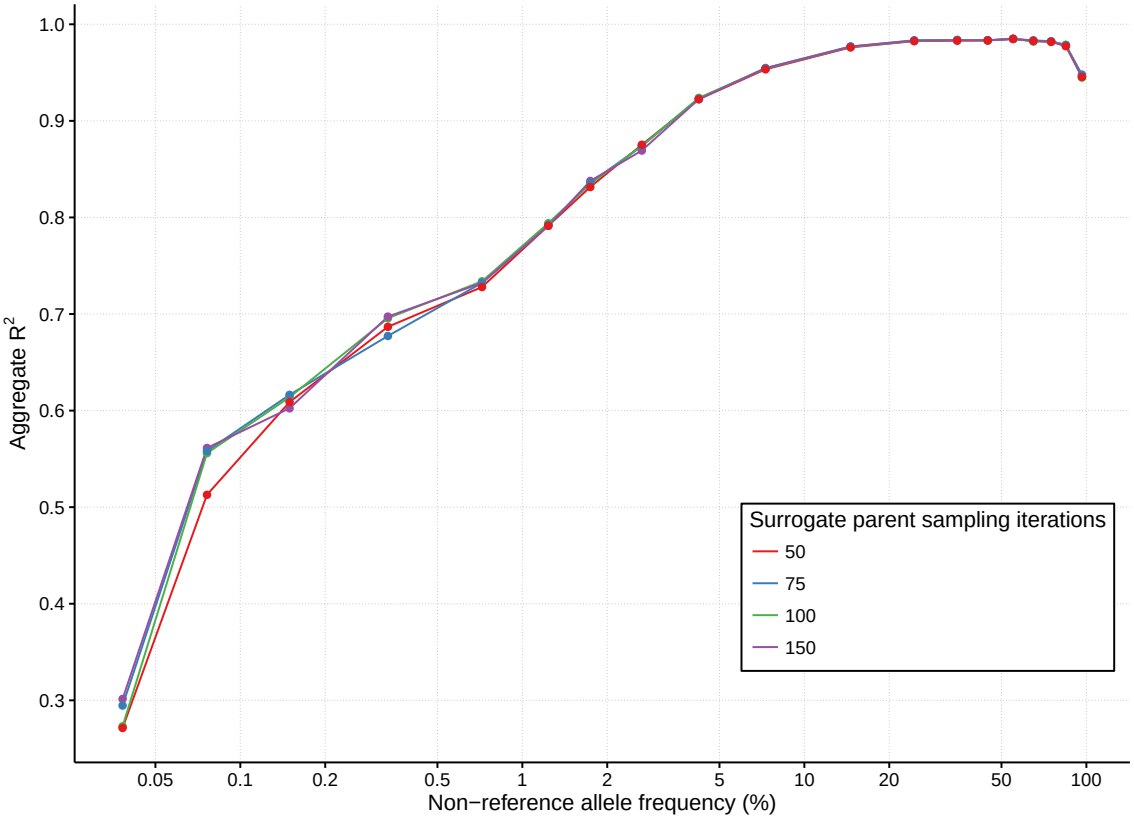

Figure 3.8: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase while varying the number of M_2 iterations.* These haplotypes were not rephased with SHAPEIT. The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs on chromosome 20. M_2 iterations between 75 and 150 appear to result in haplotypes with similar downstream imputation accuracy. Genotype discordances of all of these panels with an OMNI 2.5M chip are listed in Table 3.5.

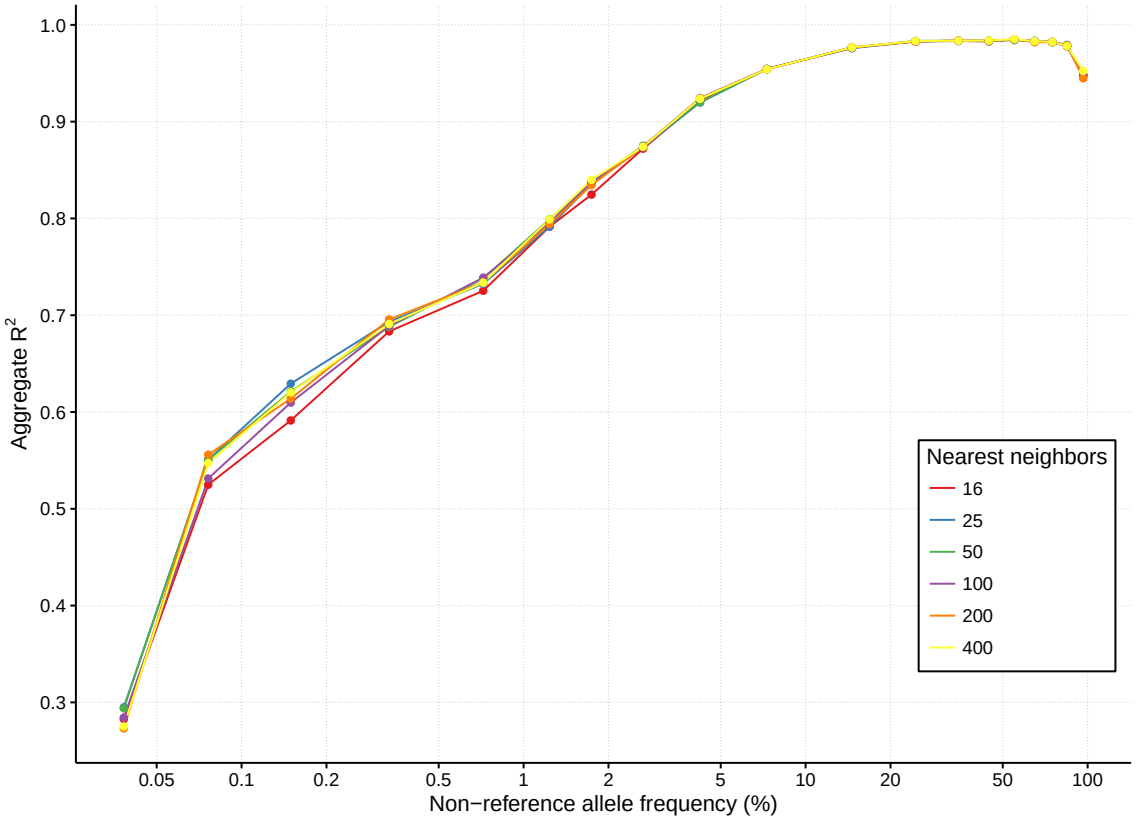

Figure 3.9: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panel phased with GLPhase while varying the number of nearest neighbors used to cluster each sample.* These haplotypes were not rephased with SHAPEIT. The HRC panel contains 32,905 mostly unrelated samples. R^2 was calculated at 175,996 bi-allelic SNPs on chromosome 20. Nearest neighbors between 25 and 400 appear to result in haplotypes with similar downstream imputation accuracy. Using 16 nearest neighbors results in a decrease in downstream imputation accuracy. Genotype discordances of all of these panels with an OMNI 2.5M chip are listed in Table 3.5.

Changes from default GLPhase	NRD	RRD	RAD	AAD
none	0.3843	0.0549	0.3365	0.2321
145 M_1 sampling iterations	0.3836	0.0547	0.3358	0.2321
70 M_1 sampling iterations	0.3867	0.0560	0.3372	0.2329
50 M_1 sampling iterations	0.3865	0.0562	0.3365	0.2330
32 M_1 sampling iterations	0.3883	0.0569	0.3378	0.2324
16 M_1 sampling iterations	0.3962	0.0599	0.3419	0.2344
150 M_2 iterations	0.3747	0.0524	0.3330	0.2225
75 M_2 iterations	0.3932	0.0567	0.3420	0.2394
50 M_2 iterations	0.4133	0.0606	0.3561	0.2537
400 nearest neighbors	0.4109	0.0589	0.3603	0.2467
100 nearest neighbors	0.3644	0.0526	0.3174	0.2207
50 nearest neighbors	0.3511	0.0519	0.3021	0.2142
25 nearest neighbors	0.3489	0.0526	0.2961	0.2156
16 nearest neighbors	0.3549	0.0548	0.2984	0.2188
40 M_1 non-SA burn-in iterations	0.3844	0.0548	0.3375	0.2310
20 M_1 non-SA burn-in iterations	0.3849	0.0549	0.3376	0.2320
10 M_1 non-SA burn-in iterations	0.3847	0.0550	0.3373	0.2316
2 M_1 non-SA burn-in iterations	0.3857	0.0550	0.3388	0.2316

Table 3.5: *Genotype discordance (in percent) between OMNI 2.5M genotypes available for 364 European samples from the 1000 Genomes project and haplotype panels shown in figures 3.7, 3.8, and 3.9.* Discordances were measured at 42,229 SNPs on chromosome 20. The columns labeled "NRD", "RRD", "RAD", and "AAD" respectively contain the non-reference, homozygous reference allele, heterozygous, and homozygous alternate allele genotype discordances. "none" represents GLPhase v1.4.13 using pre-existing haplotypes, 95 M_1 sampling iterations, five M_1 non-SA burn-in iterations, 100 M_2 iterations, and 200 nearest neighbors. M_1 non-SA burn-in iterations, M_2 iterations, and the number of nearest neighbors have a measurable impact on genotyping accuracy.

effect of nearest neighbors will depend on the quality of pre-existing haplotypes. The 1000 Genomes samples used for genotype accuracy testing have relatively good pre-existing haplotypes, so it is not surprising that a smaller number of nearest neighbors would be sufficient to phase these samples accurately.

3.6 Implementation changes to run on Graphics Processing Unit

Benchmarking of the SNPTools code indicated that SNPTools spends almost all of it's time computing likelihoods for different proposals of S as described in case 1 of subsection 3.1.4. We explored speeding up that calculation using a Graphics Processing Unit (GPU).

GPUs rely on the single instruction multiple data (SIMD) paradigm. In contrast to most desktop computers, GPUs take a single instruction and apply it to an array of data in parallel. If a problem can be formulated as applying a function to each element of an array of data, then a GPU can process the transformation of each data point in parallel using multiple thousands of cores. However, these cores are very simple, and special care needs to be taken to use them correctly. This leads to some technical limits to parallelizing a function on a GPU related to the speed with which new data can be made available to a core, and the limited amount of storage in the form of registers that are available to each core. This means that the temporary data needed by the parallelized function has to be small, and that the ratio of instructions performed per byte of data by the function should be large. Experts in the field expect algorithms to have a ratio of at least 20 instructions per byte of data in order to be promising for running on a GPU.

The SNPTools algorithm exhibits the kind of parallelism suitable for a GPU. During a M_1 iteration, the search for an individual's surrogate parents in H is performed

for each individual independently. This means that the surrogate parent search involving repeat calls to the case 1 HMM (see subsection 3.1.4) can be parallelized across individuals. In addition, the SNPTools HMM is small because it only contains four states. This allowed me to implement the case 1 HMM using just over 70 registers, well below the technical limit of 255 registers on the GPU available for testing (NVIDIA Tesla K20m).

The issue of data input size also had to be addressed. For every GL, the HMM performs about six instructions. As every GL is stored as a four byte float, this leads to a ratio of 1.5 instructions per byte of data. However, this ratio can be improved by compressing the GLs. After talking to Richard Durbin, who suggested trying vector quantization (Gray, 1984), plots of normalized GLs of the HRC pilot cohort in randomly selected regions of chromosome 20 were inspected (data not shown). It was discovered that the GLs do appear to cluster quite tightly into less than 20 clusters. Consequently, functionality for compressing the GLs using vector quantization was implemented in the following way:

1. The triplet of GLs for every site and individual are normalized such that they sum to 1.

2. Lloyd's algorithm (Lloyd, 1982) is used to fit 16 clusters to the first and second GL of every normalized GL triplet. If a cluster centroid does not have any points associated to it, then the centroid is assigned to a random GL, and clustering is continued.

3. The 16 clusters are stored in a code book, and every triplet of GLs is replaced by four bits corresponding to the cluster the triplet of GLs is closest to.

This represents a 24-fold compression that brings the ratio of instructions per byte from 1.5 to 36, although this does not count an overhead of a few extra instructions for decoding the GLs on the GPU.

An example of this clustering scheme applied to an example region of the HRC GL data can be seen in Figure 3.10. This region corresponds to region 4 of the regions used in Figure 3.3. This region represents the median of clustering quality compared to the other four regions examined (see Figure B.1 and Table B.1).

To compare speedup of the GPU implementation, we called GLs on 1,024 sites from the HRC pilot project consisting of GLs of 12,753 samples (see section 4.2). Running GLPhase with pre-existing haplotypes on an Intel Xeon E5-2690 CPU running at 2.90GHz and hyperthreading enabled took 65 minutes. Using the GPU enabled version of GLPhase on the same machine together with a Tesla K20m GPU took three minutes. We conclude that the GPU implementation provides about a 20 fold speedup over the CPU-only code on the machine tested and the sample size used.

Downstream imputation accuracy of the GPU implementation compared to the CPU implementation is shown in Figure 3.2, and genotyping accuracy on European 1000 Genomes Project individuals is given in Table 3.4. It appears from Figure 3.2 that there is some loss of downstream imputation accuracy for rare variants when using the GPU implementation compared to the CPU implementation. A loss in genotyping accuracy when using the GPU implementation is also evident from Table 3.4. The only algorithmic difference between the GPU and CPU implementations is the use of vector quantization to compress GLs for the GPU. It may be that better genotyping accuracy can be obtained with a GL clustering scheme that uses more clusters or spreads the clusters evenly across the space to be quantized.

The GPU implementation was a candidate for use with the HRC (chapter 4). However, the relative lack of GPUs at the Wellcome Trust Centre for Human Genetics was a practical impediment to the use of my GPU implementation for the HRC. While the center has a cluster consisting of over 1,440 CPU cores, it only has a few GPUs. Few bioinformatic applications are written for a GPU, and this is reflected in the relative availability of CPU and GPU compute resources at the center.

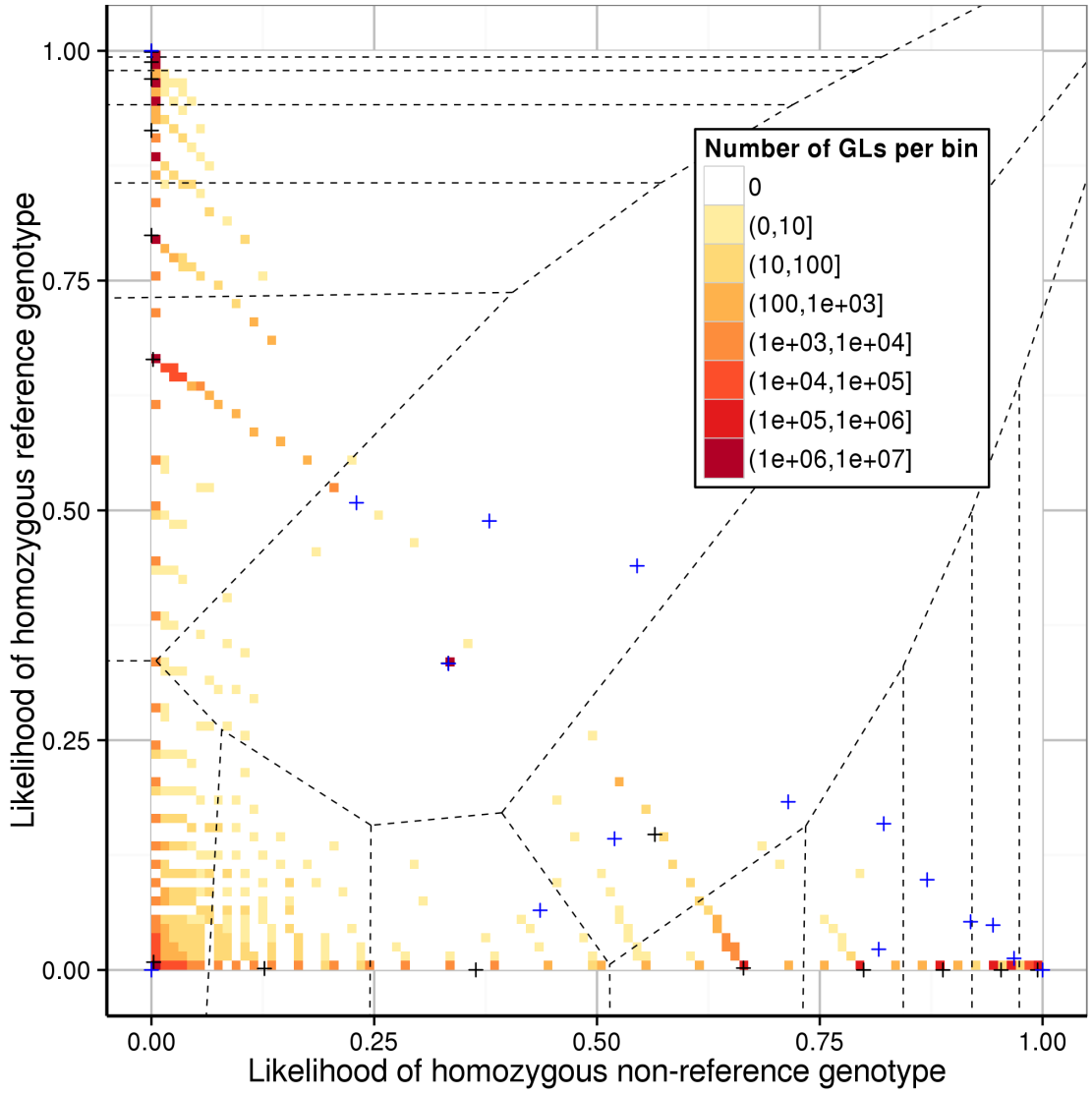

Figure 3.10: *Normalized GLs of region 4 in Table B.1 with results of k-means clustering.* GLs have been binned into a grid of 100x100 squares for presentation purposes. The number of GLs in each bin is represented as a color, which changes on a logarithmic scale. Blue crosses are initial centroids, and black crosses represent locations of centroids after clustering. Dashed lines indicate the Voronoi cells defined by post-clustering centroids. In step 3 of the GL clustering process all GLs inside a Voronoi cell are coded as the centroid of that cell.

3.7 Discussion

In this chapter changes are described that were made to the SNPTools model to adapt it to calling genotypes from GLs of tens of thousands of samples. My changes reduce complexity of the model from $O(N^2)$ to $O(N \log N)$ by incorporating pre-existing haplotypes and reference panel haplotypes into the model. We also present a GPU implementation of the SNPTools model, which reduces run time by an order of magnitude over the CPU implementation at the cost of genotyping accuracy. All of these changes are available as a binary called GLPhase.

The advantages of GLPhase are as follows:

1. GLPhase makes the unified calling of genotypes from GLs in tens of thousands of samples possible, whereas the expected run time of the original SNPTools or BEAGLE code on these data sets is impractical.

2. The GPU implementation allows for another order of magnitude speedup in computation using my model or the SNPTools model, if the sample size of the input data is sufficiently large ($>$ 10k samples).

The disadvantages of GLPhase are:

1. When calling GLs in 9,300 CONVERGE samples using the 1000 Genomes Phase 1 reference panel, GLPhase underperforms in genotyping accuracy compared to BEAGLE while keeping run time the same.

2. When pre-existing haplotypes are available for sample sizes below 10k-20k samples, BEAGLE may be a better choice because it calls better genotypes at a relatively small extra cost compared to GLPhase.

3. The GPU implementation appears to be less accurate than the CPU implementation, and more research is needed to discover the cause of the loss of accuracy.

There are several avenues that further investigation could follow. The problem of calling genotypes from GLs and phasing those genotypes might more productively be thought of as two separate problems. As is evidenced by Figure 4.12, rephasing genotypes called with my method using SHAPEIT may lead to a boost in downstream genotype imputation accuracy. My goal during development was to develop a method that produces haplotypes that maximize downstream genotype imputation accuracy. However, another metric that should have been looked at more is genotyping accuracy. It may be that trimming my method towards optimal genotyping accuracy may lead better haplotypes from SHAPEIT, and in turn better downstream imputation accuracy.

It would be nice to be able to speed up SNPTools by performing a k-NN search on unphased GLs. One approach might be to call GLs using a flat prior or a prior derived from expectation maximization of GLs, and then define nearest neighbors through sharing of rare variants.

Another question of interest is how calling genotypes from GLs with GLPhase using a reference panel scales with larger reference panels. Although calling genotypes from GLs in 9,300 Chinese women using GLPhase performed worse than BEAGLE, depending on the relative computational complexities of GLPhase and BEAGLE, it may be that GLPhase outperforms BEAGLE when larger population-matched reference panels are used. On the one hand, the GLPhase approach to calling GLs with a reference panel is complexity $O(N)$, but one would expect to have to increase the number of M_2 iterations as the reference panel size increases. How the increase in iterations depends on reference panel size is an open question.

Chapter 4

The Haplotype Reference Consortium data set

4.1 Introduction

Current practice in SNP chip-based genome-wide association studies (GWAS) is to impute missing variants from a whole genome sequencing-based haplotype reference panel such as the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2010, 2015) using a statistical tool such as Impute2 (B. N. Howie, Donnelly, and Marchini, 2009). Marchini and B. Howie (2010) showed that this approach increases the power of a GWAS to detect genetic associations.

Recently, whole genome sequencing has become cheap enough for single groups and smaller consortia to sequence their own cohorts (e.g., Genome of the Netherlands Consortium, 2014; CONVERGE Consortium, 2015). The Haplotype Reference Consortium (HRC) was able to find 20 projects independently sequencing European genomes (see Table 4.1). However, there is evidence that imputation methods benefit from increased reference panel sample size (B. Howie, Marchini, et al., 2011). This means that imputing genotypes using all 20 reference panels may lead to increased

genotype imputation accuracy, which in turn would lead to increased power for GWAS to detect genetic associations. Unfortunately, there is no method of genotype imputation that works with more than two reference panels. An alternative approach is to create a unified reference panel from the data of these 20 projects.

The Haplotype Reference Consortium (HRC) is a consortium of groups that own whole genome sequences of humans. The analysis group of the HRC consists of six people: Shane McCarthy, Sayantan Das, Warren Kretzschmar, Richard Durbin, Gonçalo Abecasis, and Jonathan Marchini. This chapter includes work done by some of these people, and they are named in each section that their work is described in.

The aim of the first release of the HRC and of this chapter is to combine the sequencing data of all of these groups to create a unified set of predominantly European haplotypes at a high quality set of SNPs that can be used for phasing and imputation of genotypes. This work has also been published as a pre-print as McCarthy et al. (2015).

4.2 Materials and methods

Aligned Illumina sequencing reads of 38,821 humans at a depth between 4x and 45x were obtained from 20 projects listed in table 4.1. For the first release, the HRC only included sequencing data from Europeans, although an exception was made for 1000 Genomes Phase 3 samples (The 1000 Genomes Project Consortium, 2010) to make it easy for researchers to switch from using a 1000 Genomes reference panel to using the HRC panel. For each project, the groups working on them also shared initial haplotype calls made only using sequencing data from the samples that were part of the project's cohort. The initial haplotype calls were merged to create an integrated site list. The site list was used to identify variants at which to call genotype likelihoods (GLs), and to identify samples that were duplicated within and across cohorts. The

site list was used to call GLs, which were used to call and phase genotypes using the method described in chapter 3. Finally, these haplotypes were rephased using SHAPEIT to increase downstream genotype imputation accuracy.

Early in the project, Olivier Delaneau created a pilot reference panel of cohorts that were available to the project early on for testing purposes. This panel contains 13,309 samples from the cohorts listed in Table 4.1 as contributing samples in the "Pilot" column, and sites that have a non-reference allele count (AC) > 10 in the pre-existing haplotypes for those samples on chromosome 20. Unfortunately, due to overlap in project sample IDs, it is impossible to unambiguously work out which of the 18,897 samples listed in Table 4.1 contributed to that panel. Following best practices from The 1000 Genomes Project Consortium (2012), Olivier called genotypes with BEAGLE and rephased the haplotypes with SHAPEIT2. This data set will be referred to as "HRC pilot AC10", or "the pilot data".

4.2.1 Selection of variants

4.2.1.1 MAC2 and MAC5 site list creation

Every project provided us with their most recent version of their haplotypes in VCF format with one VCF for every autosome. In this chapter these haplotypes are referred to as "pre-existing haplotypes". Unless otherwise stated, Bcftools 0.2.0-rc12 was used to perform the operations described in this paragraph. An entire-autosome, SNP-only site list with alternate and total allele count information was created from the per-chromosome pre-existing haplotypes. Multi-allelic SNPs were then broken into bi-allelics[1], and these per-cohort site lists were merged into a single file using a Perl

[1] Breaking of multi-allelic variants into bi-allelic variants is a process by which a multi-allelic VCF record is mapped to N bi-allelic VCF records, where N is the number of alternate alleles in the multi-allelic VCF record. Each bi-allelic record has the reference allele of the multi-allelic record as its reference allele.

Let us assume that a multi-allelic genotype contains two alleles. If a multi-allelic genotype contains the n^{th} allele, where $n \in N$, then the n^{th} bi-allelic record will have 1, and all other bi-allelic records

Project tag	Pilot	R1	Depth	Citation (where available)
1000G	2,526	2,495	4x/Exome	The 1000 Genomes Project Consortium (2015)
AMD	3,310	3,305	4x	
BRIDGES	0	2,487	6-8x (12x)	
GECCO	0	1,131	4-6x	
GONL	748	748	12x	Genome of the Netherlands Consortium (2014)
GOT2D	2,746	2,710	4x/Exome	
GPC	0	697	30x	
HELIC	0	247	4x (1x)	
HUNT	0	1,023	4x	
IBD	0	4,478	4x + 2x	
INCHIANTI	0	676	7x	
FVG	0	250	4-10x	
VBSEQ	0	225	6x	
MTFS	0	1,325	10x	
NEPTUNE	0	403	4x	
ORCADES	399	398	4x	
PROJECTMINE	0	935	45x	
SARDINIA	3,448	3,445	4x	
FINLAND	1,940	1,918	4x	
UK10K	3,780	3,715	6.5x	McCarthy et al. (2015)
Totals	**18,897**	**32,611**		

Table 4.1: *List of projects contributing sequencing data to the HRC.* "Pilot" lists the number of samples from each cohort included in the "pilot cohorts" data set. "R1" lists the number of samples from each project included in release 1 of the HRC.

script that correctly merges alternate and total allele counts. Finally, sites with minor allele counts less than 2 and 5 were filtered out to create the "MAC2" and "MAC5" site lists, respectively.

4.2.1.2 Variant filtering based on cohort call rate

Sayantan Das, under the direction of Gonçalo Abecasis, developed an ad-hoc method for identifying sites that had been filtered out "quite often" by submitting projects. For every project, variants were called from the GLs (see section 4.2.3) available for that project at every site reported by any project. Any sites exhibiting evidence of more than one allele were marked as "called" in that project, and "not called" otherwise. Furthermore, any sites not in the project's pre-existing haplotypes (see section 4.2.4) were considered to have been filtered by that project's analysis pipeline. Sayantan then binned all variants according to how many projects called the variant and how many projects filtered out the variant. Finally, the SNP transition to transversion ration was calculated for every bin. SNPs were removed that fell into bins with a Ts/Tv ratio less than 1.7 or that had been filtered out by more than four studies (see subsubsection 4.3.1.1).

---

will have 0 at that position. The same is true for the second allele of that multi-allelic genotype. If a multi-allelic genotype has the reference allele at a position, then all bi-allelic records will have 0 at that position. For example, the multi-allelic VCF record

```
20 1 . A G,T . . . GT 1|2 0|2
```

is broken into the bi-allelic records

```
20 1 . A G . . . GT 1|0 0|0
20 1 . A T . . . GT 0|1 0|1
```

This mapping is an injection, as not all sets of diploid bi-allelic records that are valid diploid bi-allelic VCF records can be mapped to a meaningful diploid multi-allelic record. For example, it is unclear what the diploid genotype of the sample with the following VCF records is:

```
20 1 . A G . . . GT 1|1
20 1 . A T . . . GT 0|1
```

In the remainder of this chapter we will use "cohort call rate filter" and "Santy's filter" interchangeably to refer to this filtering step.

4.2.1.3 Variant filtering based on cohort statistics

Shane McCarthy annotated the MAC2 site list with cohort-based summary statistics for my analysis. VCF header was removed from the annotated MAC2 site list and the data was loaded into R.

Based on other work done by Shane McCarthy (data not shown), we know that the GPC haplotypes have a high genotyping error rate despite the high sequencing coverage of the GPC project. These findings were also made in quality control plots (see results presented in subsection 4.3.1). For this reason, the GPC cohort summary information was excluded from this part of the site calling process.

We filtered sites for which any of the following statements were true:

1. Minor allele count less than five
2. Per-cohort Hardy-Weinberg Equilibrium (HWE) p-value $< 10^{10}$ and cohort is not 1000 Genomes
3. Overall inbreeding coefficient (ICF[2]) less than -0.1
4. Minor allele frequency (MAF) > 0.1, site called in less than three cohorts, and site not called in 1000 Genomes

[2]The site-wise estimate of the inbreeding coefficient F (ICF) used by `vcf-annotate` is defined as

$$\begin{aligned} \text{ICF} &= 1 - f_o/f_e \\ f_o &= 2 \cdot n_{\text{het}}/n_{\text{all}} \\ f_e &= 2 \cdot (1 - n_{\text{alt}}/n_{\text{all}}) \cdot n_{\text{alt}}/n_{\text{all}}, \end{aligned} \tag{4.1}$$

where f_o is the observed number of heterozygous genotypes at a site, f_e is the number of heterozygous genotypes expected under Hardy-Weinberg equilibrium. n_{het}, n_{all}, and n_{alt} are the number of heterozygous genotypes, number of alleles, and number of alternate alleles at a site, respectively. A site that has fewer heterozygotes than expected will have an ICF close to 1. A site that has as many heterozygotes as expected will have an ICF close to 0. Finally, a site that has more heterozygotes than expected will have an ICF close to -1. Inbred populations will tend to have an ICF value closer to 1, as inbreeding reduces the number of heterozygotes in a population.

5. Site only called in GONL or CROHNS

The evidence leading to our choice of these filters is presented in subsubsection 4.3.1.2.

This site list contained 46,852,018 sites. The remaining sites were then intersected with a site list derived by filtering across cohorts that was created by Sayantan Das for a site list comprised of 44,038,997 sites. Finally, 4,914,335 sites found on a selection of common SNP genotyping arrays were added back in. Sayantan Das prepared the list of SNPs found on common genotyping arrays. The final site list contains 44,187,567 sites.

In the remainder of this chapter we will use "cohort statistic-based filter" and "Winni's filter" interchangeably to refer to this filtering step.

4.2.2 Sample filtering

Before phasing, the consortium removed 9 samples that were used to evaluate downstream genotype imputation accuracy, 261 samples that were genetic duplicates of other samples in the data set, 31 1000 Genomes Phase 3 related samples, and 8 samples with labeling inconsistencies to arrive at a total of 32,611 samples (see Table 4.1).

After phasing of the reference panel 83 individuals were removed because they had not given consent to be included in the reference panel. Shane McCarthy also repeated the duplicate detection process on the final HRC genotype calls, since some studies increased in size late on within the analysis process. This resulted in an extra 40 samples being removed and a total of 32,488 samples in the first release of HRC.

4.2.2.1 Testing samples

Nine of the ten samples listed in section 4.2.7.1 that were used for downstream imputation accuracy assessment were also part of the 1000 Genomes Phase 3 cohort. We removed these samples from all haplotype reference panels in order to be able to assess downstream imputation accuracy of the panels in a representative manner.

4.2.2.2 Duplicate detection

All work in this section was performed by Shane McCarthy. Some of the cohorts in the HRC contain twins or samples that are also part of other cohorts. The approach taken by Shane to detect these "genetic duplicates" was to select a small number of well genotyped sites at which to do this comparison. For a site to be included it had to:

1. be bi-allelic
2. have a European minor allele frequency $> 5\%$ in 1000 Genomes Phase 3
3. have no missing data in any of the cohorts

Of these sites, 1000 were selected at random. For every pair of samples, the number of genotypes that differ at these sites was calculated using the Bcftools subcommand `gtcheck`. Visual inspection of a histogram of the number of differences at these sites with any other samples revealed a clear set of 269 sample pairs with very few genotype differences (all samples to the left of the red vertical line in Figure 4.1). Some of the samples were duplicated more than once, resulting in only 261 samples being removed.

4.2.3 Calling of genotype likelihoods

The genotype likelihood (GL) calling procedure was designed by Shane McCarthy to run on aligned reads submitted to the HRC analysis group by the projects. The majority of the read data were stored at the Wellcome Trust Sanger Institute and the University of Michigan compute centers. The ProjectMINE read data were stored at the Wellcome Trust Centre for Human Genetics. Shane, Sayantan Das, and I split up the cohorts to run the procedure on among us. I only ran the procedure on the ProjectMINE data. Shane merged the GLs created in each of the three centers. The procedure for calling genotype likelihoods is described in detail in appendix A.

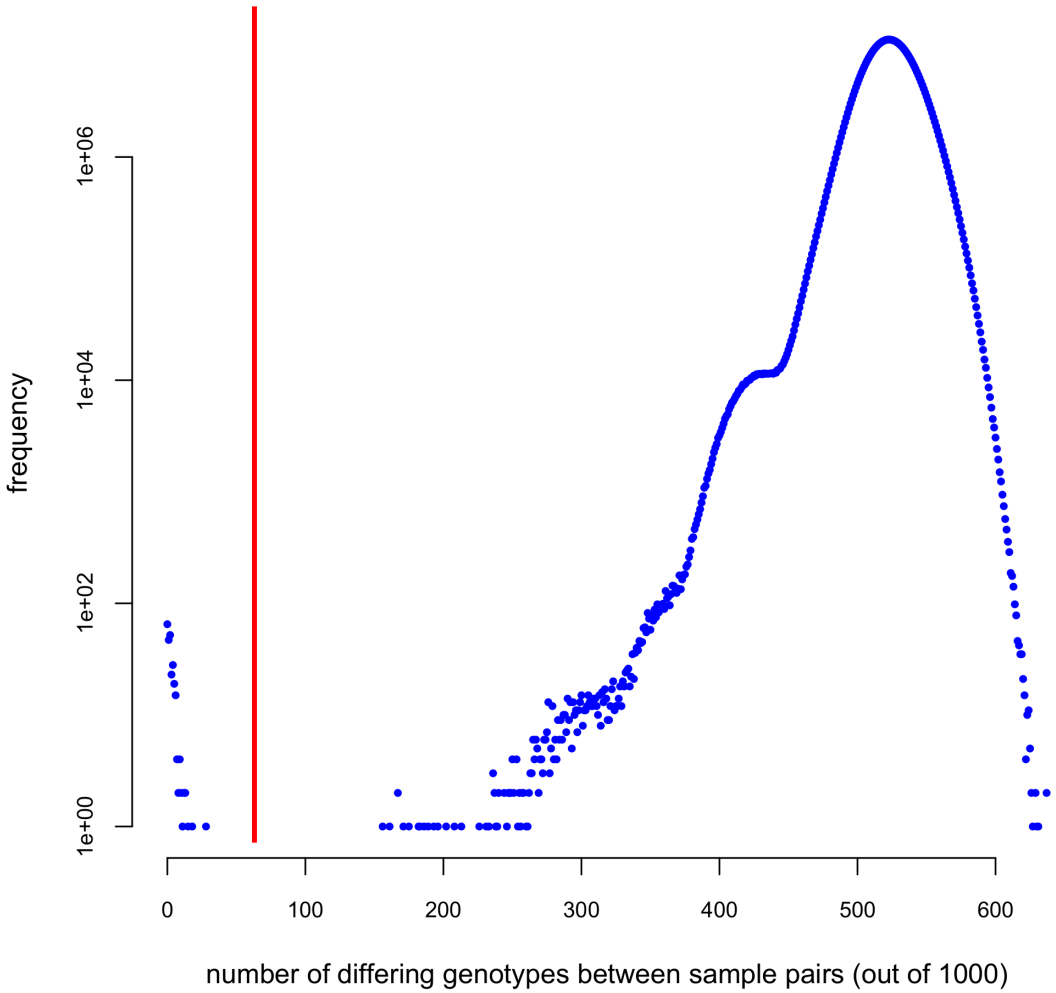

Figure 4.1: *Histogram of the number of differences between all pairs of samples.* Differences were counted at 1000 easy to genotype sites on chromosome 20 of the submitted haplotypes. All samples to the left of the red vertical line were excluded from further analysis. This figure was created by Shane McCarthy.

4.2.4 Merging pre-existing haplotypes - Missing set to major

The merged pre-exsting haplotypes described in this section were used in order to reduce the number of iterations GLPhase had to be run to impute genotypes. Bcftools v1.1 was used for all operations unless stated otherwise. We processed the cohort chromosome VCFs in the following way:

1. Only SNPs were kept. This is not a strict requirement of GLPhase. However, only SNPs were used because few cohorts had submitted indels, and as indels were not the focus of the HRC, no proper quality control had been performed on indels.

2. Filtered samples were removed.

3. Multi-Allelic SNPs were broken into bi-allelics.

4. An in-house Perl script was used to randomly assign phase to unphased genotypes at sites with a minor allele count of 1. Many of the pre-existing haplotypes were phased using Beagle, which does not phase singletons. For the purpose of clustering in GLPhase, we preferred to have poorly phased rare variants over not having them at all.

5. VCFs created at the University of Michigan had their sample IDs replaced with a unique ID using an in-house Perl script.

6. All VCFs were intersected with the MAC5 site list.

For every chromosome, we then merged all the cohort VCFs into one. These VCFs were then processed in the following way:

1. Sites containing an unphased, non-missing genotype were removed using GNU Grep because it was unclear why these genotypes were unphased.

2. For all sites, missing genotypes were replaced by a phased genotype that is homozygous for the site's major allele using a Bcftools plugin called "missing2ref". GLPhase does not accept missing alleles for pre-existing haplotypes, and this approach appears to work well for the HRC. However, it is possible that keeping these genotypes as missing and changing the distance calculation in GLPhase to handle missing alleles might improve GLPhase clustering.

4.2.5 Calling and phasing of genotypes

4.2.5.1 k-NN search on pre-existing haplotypes

We phased and imputed haplotypes for the HRC using GLPhase v1.4.13 while taking advantage of pre-existing haplotypes (see section 3.3) and after introducing a genetic map (see section 3.4).

4.2.6 Rephasing of haplotypes with SHAPEIT

Using a set of instructions provided by me, Sayantan Das and Shane McCarthy rephased the haplotypes derived from GLPhase using a new version of SHAPEIT (O'Connell, Sharp, et al., 2016) that uses a partitioning algorithm for finding haplotype nearest neighbors that has better complexity than the method used for nearest neighbor search in SHAPEIT2. At our sample size, this reduced runtime approximately four fold. Chromosomes were phased in chunks consisting of 16,000 variants plus 3,300 variants overlapping with neighboring chunks on either side. The non-default command line option `-w 0.5` was used for SHAPEIT3. Chunks were ligated using ligateHAPLOTYPES (available on the Impute2 website). SHAPEIT does not handle multiple variants at the same genomic coordinate, so multi-allelic sites were shifted by one or two positions for rephasing, and then moved back to their original position after chunk ligation.

4.2.7 Downstream genotype imputation accuracy assessment

4.2.7.1 Complete Genomics data preparation

Complete Genomics called genotypes from high coverage short read sequencing of around 300 samples available to the 1000 Genomes Project Phase 3 for genotyping quality assessment. The HRC also obtained these data and used ten randomly selected CEU Europeans for downstream imputation accuracy assessment. These are the sample IDs of the ten samples we used for downstream imputation accuracy assessment: NA07346, NA11832, NA12005, NA12156, NA12414, NA12760, NA12762, NA12776, NA12813, NA12828. These samples were removed from the HRC data set before genotype imputation as described in section 4.2.2.1.

A pseudo-GWAS panel was created by filtering out genotypes at all sites that did not have positions in the Illumina 1M genotyping chip site list or were not in the HRC pilot sites. Genotypes at the filtered out sites were kept to create a validation data set if those sites had no missing genotypes in the ten European samples.

For Figure 4.13, a pseudo-GWAS panel was used that was not filtered on the HRC pilot sites.

4.2.7.2 Genotype imputation into pseudo-GWAS panel

Genotypes were imputed from a haplotype panel of interest into the pseudo-GWAS genotypes using Impute2 (B. N. Howie, Donnelly, and Marchini, 2009). The non-standard options used with Impute2 were `-Ne 15000 -k_hap 1000`. Imputation was performed in chunks between two and six mega bases in size.

4.2.7.3 Genotype imputation accuracy calculation

Genotype imputation accuracy into the pseudo-GWAS set was evaluated at the intersection of the sites in the validation data, the haplotype reference panel being

evaluated, the 1000 Genomes Phase 1 SNPs, and the sites in the HRC pilot data. The sites were binned by non-reference allele frequency calculated by expectation maximization of GLs in the HRC pilot data Europeans. For each allele frequency bin, aggregate R^2 was calculated across all sites and all samples.

For Figure 4.13, allele frequencies were estimated by expectation maximization from GLs of the European HRC samples of the full HRC set (32,920 - 2001 = 30919 samples).

4.3 Results

4.3.1 Selection of variants

4.3.1.1 Cohort call rate-based variant filtering

Figure 4.2 shows work done by Sayantan Das to illustrate the distribution of SNPs across chromosome 20 stratified by how many cohorts called and filtered out each SNP. SNP density is indicated by black lines. The centromere and telomeric region of chromosome 20 are delineated by three vertical black lines. These regions tend to be more repetitive, and this makes calling variants harder. In an ideal situation, the SNP density in these regions would be no different from the rest of the chromosome. In Figure 4.2 we can see that this appears to be true for variants that were filtered out no more than five times.

Figure 4.3 shows work done by Sayantan Das to quantify the quality of SNPs as a function of how many cohorts called a SNP and then filtered it out again. The quality metric displayed in the figure is transition to transversion (Ts/Tv) ratio. Based on inter-species comparisons and multiple human sequencing projects, expected values for Ts/Tv are around 2.0-2.1 for genome-wide data sets and 3.0-3.3 for exonic variation (DePristo et al., 2011). Lower numbers tend to indicate enrichment for false SNP calls.

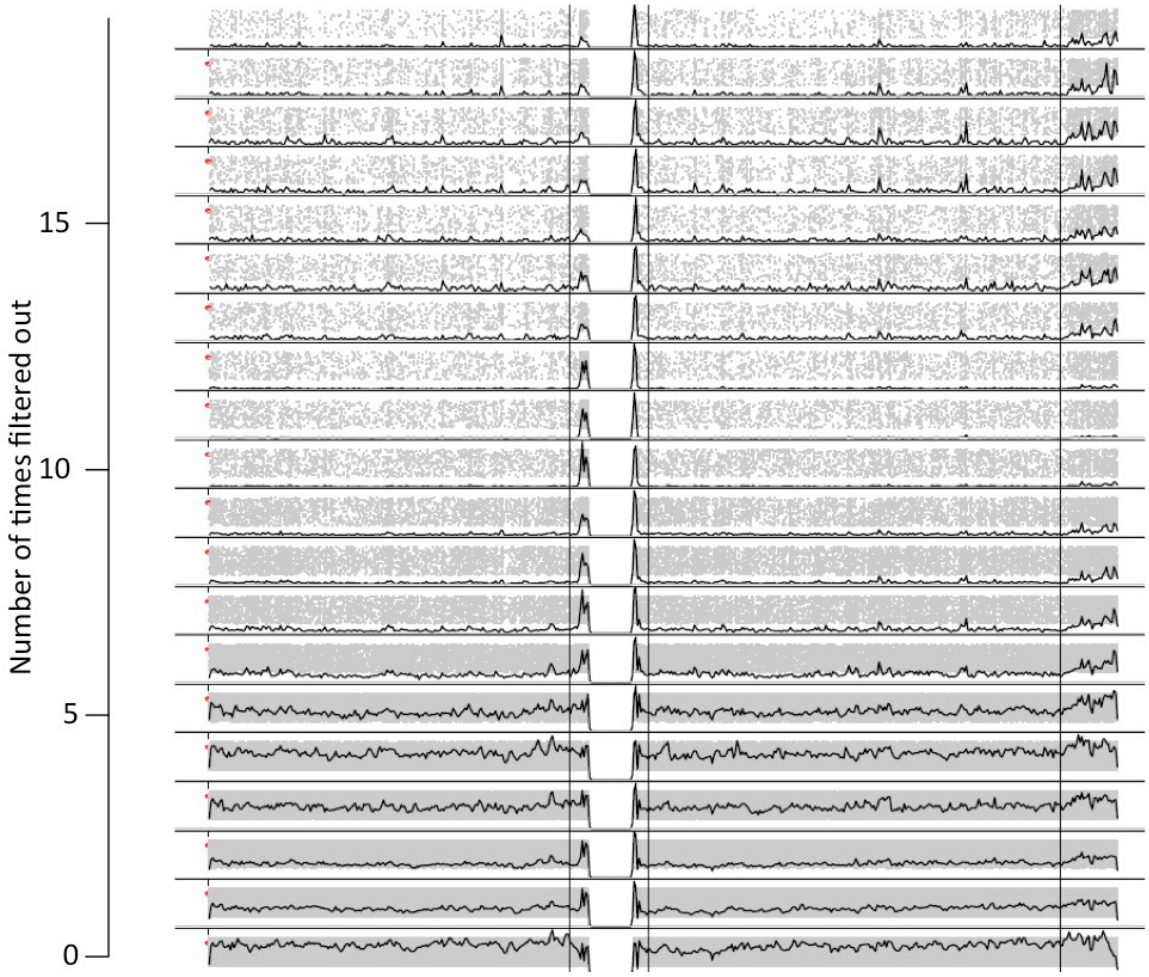

Figure 4.2: *Distribution of variants across chromosome 20 stratified by number of cohorts that filtered out the variant.* This figure was prepared by Sayantan Das at the University of Michigan. The centromere (center) and and telomere (right) are delineated by vertical black lines. The centromere and telomere are enriched for variants that have been filtered by more than five cohorts.

Sites that were only filtered out a hand full of times have Ts/Tv ratios that are higher compared to SNPs that were filtered out by more cohorts. Sites that were filtered out by every cohort that called them generally have very low values of Ts/Tv.

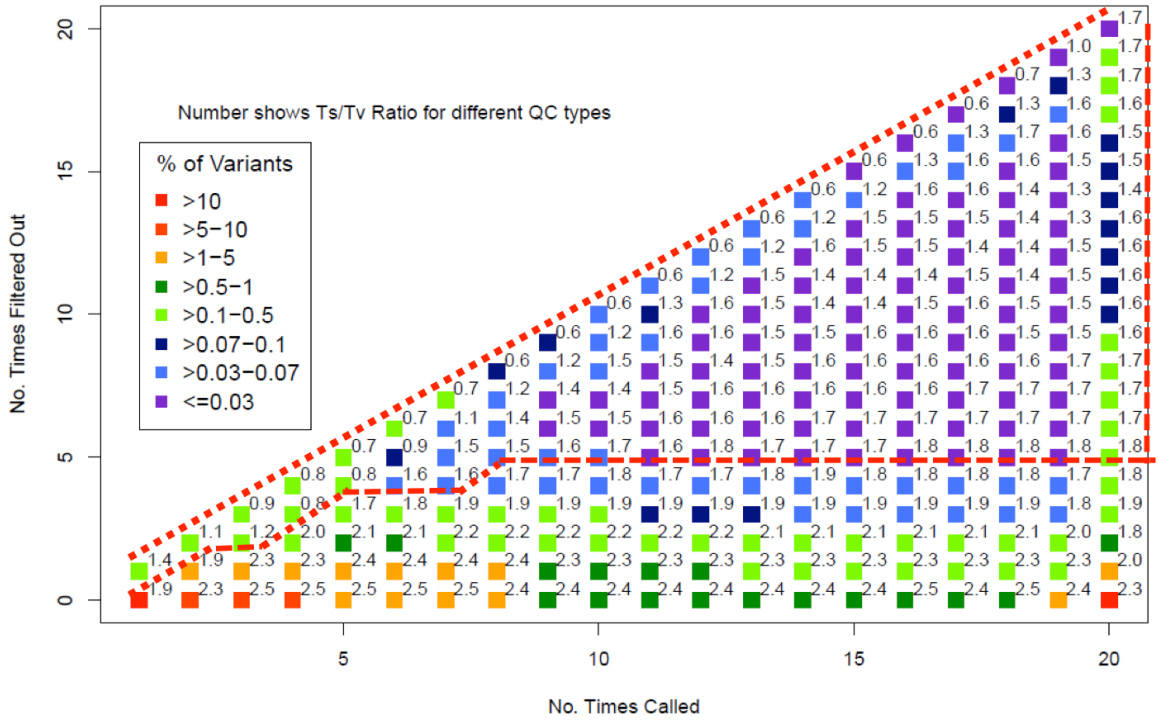

Figure 4.3: *Variants binned by number of times called and number of times filtered out.* This figure was prepared by Sayantan Das at the University of Michigan. Bins above the lower and below the upper dashed red lines were excluded. Each square and number represents a bin. The number is the transition to transversion ratio of the SNPs in the bin. The color of the bin indicates what percentage of variants fall in that bin (see legend).

Based on figures 4.2 and 4.3, the HRC analysis group decided to remove sites that were filtered out by more than five cohorts, and bins that had a Ts/Tv ratio below 1.6. The polygon delineated by the dashed red lines in figure 4.3 was set by the HRC analysis group to follow these two requirements, while only filtering out a small number (3.7 million, 7.7%) of SNPs. The polygon is the set of sites that the HRC analysis group filtered out in this way.

4.3.1.2 Variant filtering based on cohort statistics

My investigation of SNP call filtering based on cohort statistics went through several iterations that included feedback from the HRC analysis group. However, from the start we focused on the metrics we knew to be important for variant quality control: minor allele count (MAC), number of cohorts that called a variant (NC), Hardy-Weinberg equilibrium (HWE) p-value, per-site inbreeding coefficient (ICF), and minor allele frequency (MAF).

An important type of site we wanted to filter is a SNP that is only called in one or two cohorts, but has a high allele frequency in those cohorts. As all cohorts, except for 1000 Genomes Phase 3, are European samples, allele frequencies of SNPs in those cohorts should be similar. Therefore, we would expect SNPs with a high MAF in one European-only cohort to also have a high MAF in other European-only cohorts. In figures 4.4 and 4.5 we plot the $\log_{10}$ HWE p-values of SNPs that were only seen in one or two cohorts, respectively, and that had a high MAF. Having a large number of SNPs in this category or having a large proportion of SNPs with low $\log_{10}$ HWE p-values are signs of poor SNP calling pipelines that have a large number of false SNP calls in cohorts that have no reason to be deviating from HWE. The GPC cohort consists of samples from people with schizophrenia or bipolar disorder from the general population of the USA. As such one would not expect this cohort to deviate from HWE. Therefore, figures 4.4 and 4.5 provide evidence that the GPC SNP calls may be of a poor quality. To a lesser extent, the GoNL SNP calls also appear to have more low HWE p-values compared to the other cohorts.

Inspection of per-cohort ICF values for SNPs on chromosome 20 called in only one cohort revealed similar issues to the analysis of common private SNPs. In cohorts without population structure or inbreeding, SNPs with high quality genotype calls should have an ICF around 0. Figure 4.6 shows GPC has many SNPs with very high and very low ICF values. GoNL also has many SNPs with negative ICF values. From

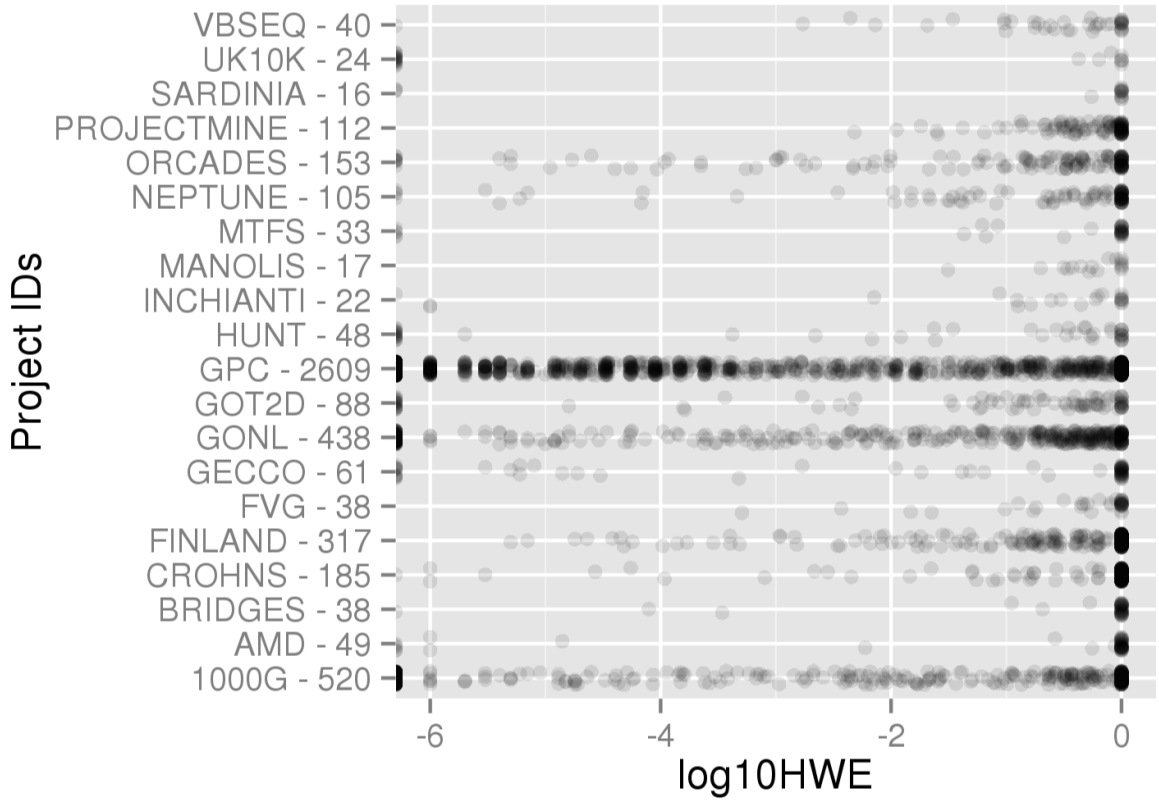

Figure 4.4: $\log_{10}$ *HWE p-value of SNPs seen only in one cohort and with a MAF* > 0.1 *on chromosome 20, stratified by cohort.* The number of sites in each cohort is given next to the cohort name. GPC has a large number of sites compared to the other cohorts, and many of the sites have very low $\log_{10}$ p-values.

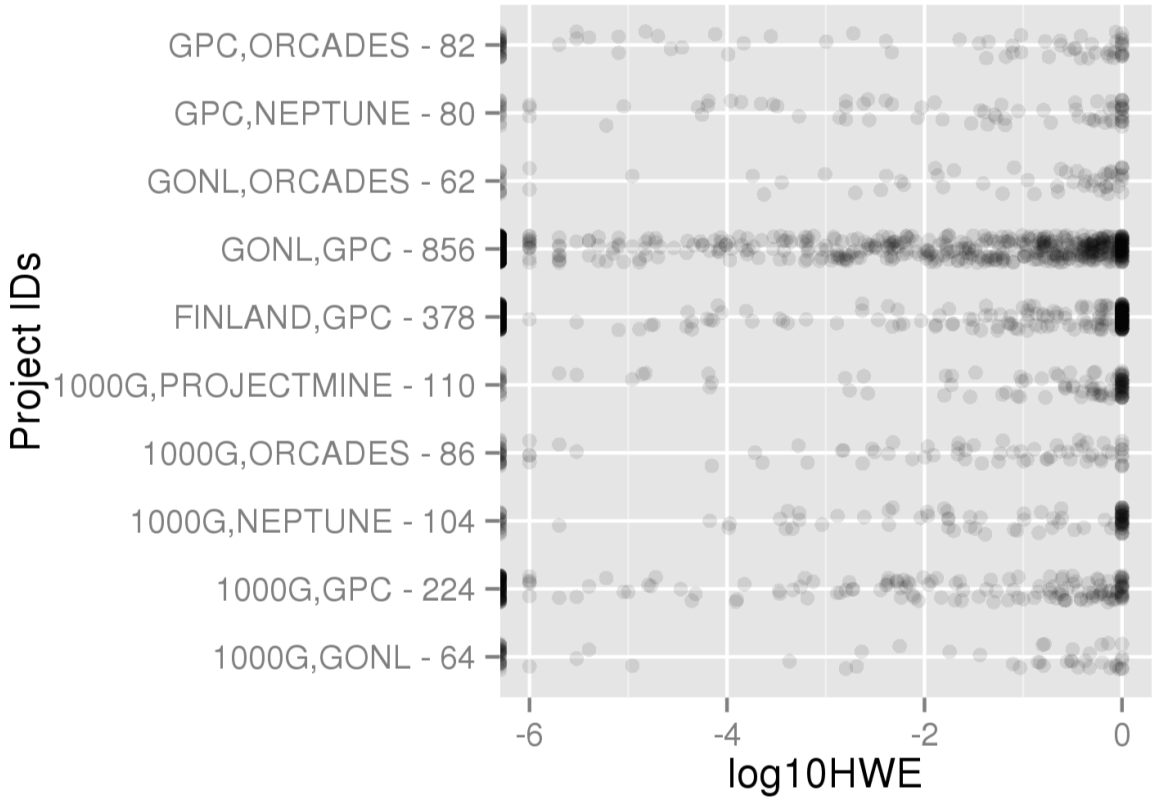

Figure 4.5: $\log_{10}$ *HWE p-value of SNPs seen in two cohorts and with a MAF* > 0.1 *on chromosome 20, stratified by cohort.* The number of sites shared by each pair of cohorts is given next to the cohort names. As in Figure 4.4, GPC is part of many cohort pairs. The large overlap of GPC and GoNL sites, and that these sites have very low $\log_{10}$ p-values indicates that variants in these cohorts may have been poorly called.

this plot it appears that most cohorts have SNPs with ICF values above -0.1. For this reason the HRC analysis group chose a per-cohort ICF value of -0.1 as a filtering cutoff for SNPs.

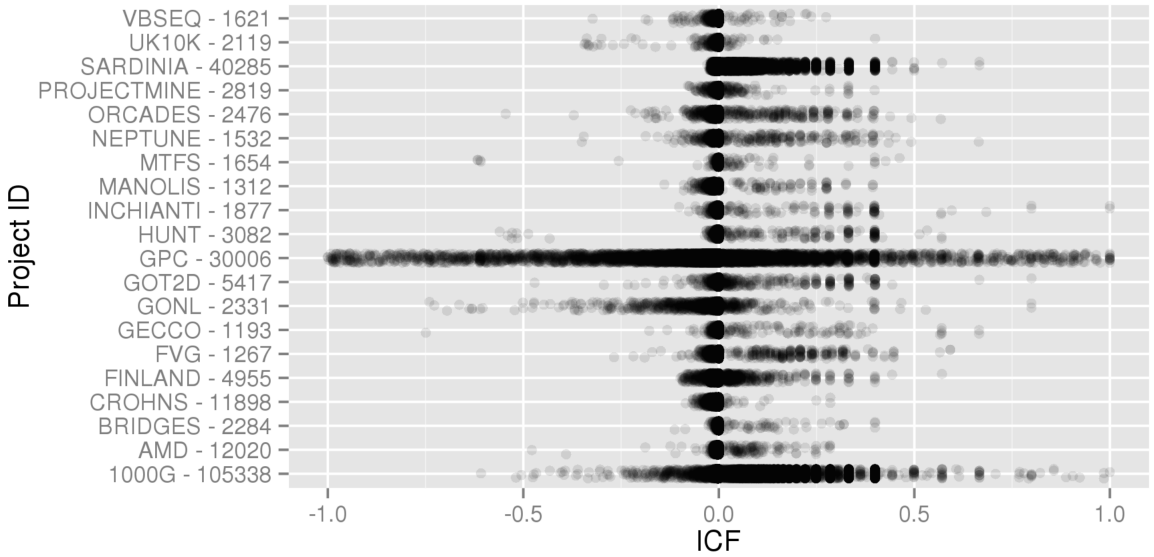

Figure 4.6: *ICF of chromosome 20 SNPs seen in only one cohort.* High quality SNPs are expected to have an ICF of around 0 in cohorts without population structure and no inbreeding. GPC has a large number of SNPs with very high and low values of ICF. GoNL also has a large number of SNPs with negative ICF values.

The Venn diagram in figure 4.7 shows by how many sites the cohort statistic-based site filters overlap. All of these filters have substantial numbers of sites that are only filtered by that filter. The filters filtering out the most sites are the filter for private GPC, CROHNS, or GoNL sites, and the filter for an ICF value below -0.1.

Table 4.2 lists the number of sites remaining after successive application of the cohort summary statistic-based site filters.

Based on figures 4.4, 4.5, 4.6, and 4.7, the HRC analysis group decided to completely exclude GPC site calling data from the variant filtering process, and to filter out SNPs that were only seen in GoNL. An earlier version of the CROHNS cohort haplotypes had had similar issues to the GoNL sites, which is why sites from the CROHNS cohort were initially filtered. After the CROHNS haplotypes were updated, the CROHNS site filter was mistakenly not removed. Figure 4.8 shows the

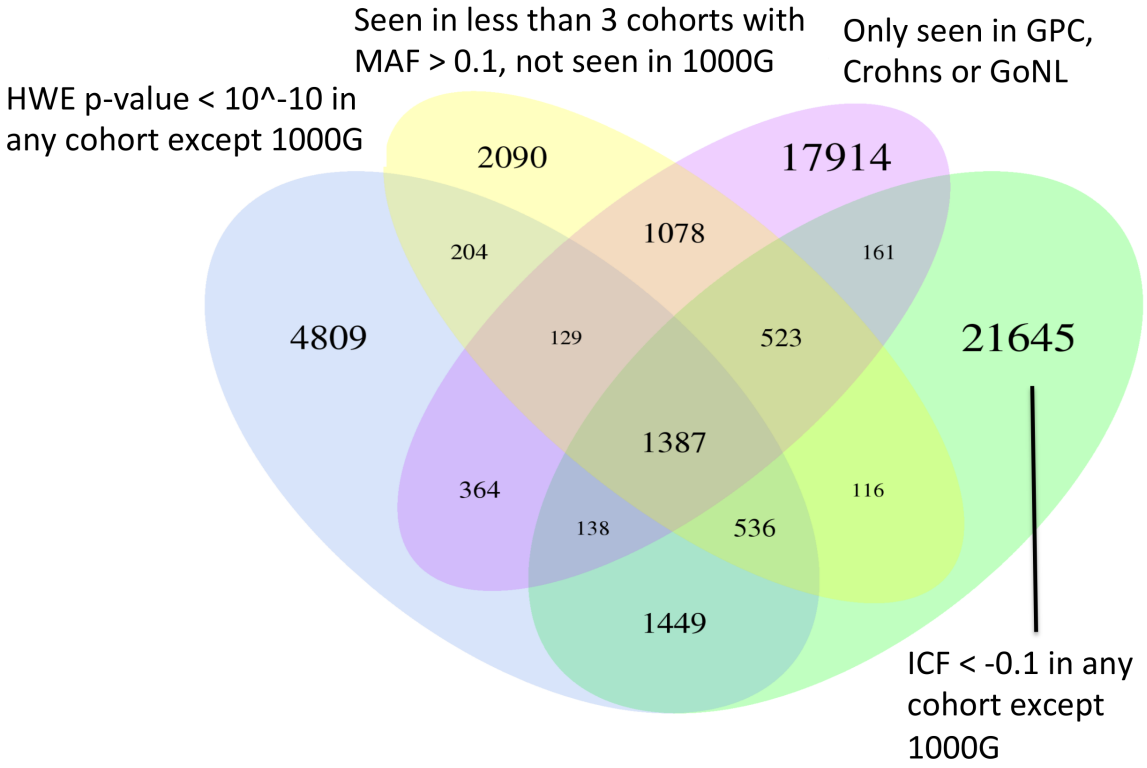

Figure 4.7: *Venn diagram of sites filtered out on chromosome 20 by cohort statistic-based filters.*

Filter	Number of sites on		% change	
	Chr. 20	Autosomes	MAC5	Santy
MAC < 2	2,169,107	96,091,440	192.10	0.00
MAC < 5	1,128,997	50,014,567	100.00	7.40
MAC < 5, no GPC data	1,064,569	47,160,407	94.29	49.60
HWE p-value < 10^-10 in any cohort except 1000G	1,061,782	47,036,943	94.05	51.60
ICF < -0.1 across all cohorts	1,061,263	47,013,951	94.00	52.10
Seen in less than 3 cohorts with MAF > 0.1 and not seen in 1000G	1,059,125	46,919,238	93.81	54.40
Only seen in Crohns or GoNL	1,051,276	46,571,527	93.12	58.20
Position in Santy's filter	1,017,422	45,071,795	90.12	100.00

Table 4.2: *Incremental application of site filters based on cohort summary statistics.* "Chr. 20" refers to chromosome 20. The numbers for the autosomes are extrapolated from the chromosome 20 numbers using an empirically obtained factor of 44.3. "% change MAC5" refers to the number of sites left after each filter compared to the number of sites left after the MAC < 5 filter. "% change Santy" refers to the proportion of sites filtered by each filter that are also filtered by Sayantan Das' cohort call rate-based filtering approach.

sites filtered by each filter, including the cohort call rate filter. Removing the GPC calls removes the majority of sites that were captured by the ICF and HWE p-value filters (compare figures 4.7 and 4.8). The majority of sites filtered out are now filtered by the cohort call rate filter (see figure 4.8).

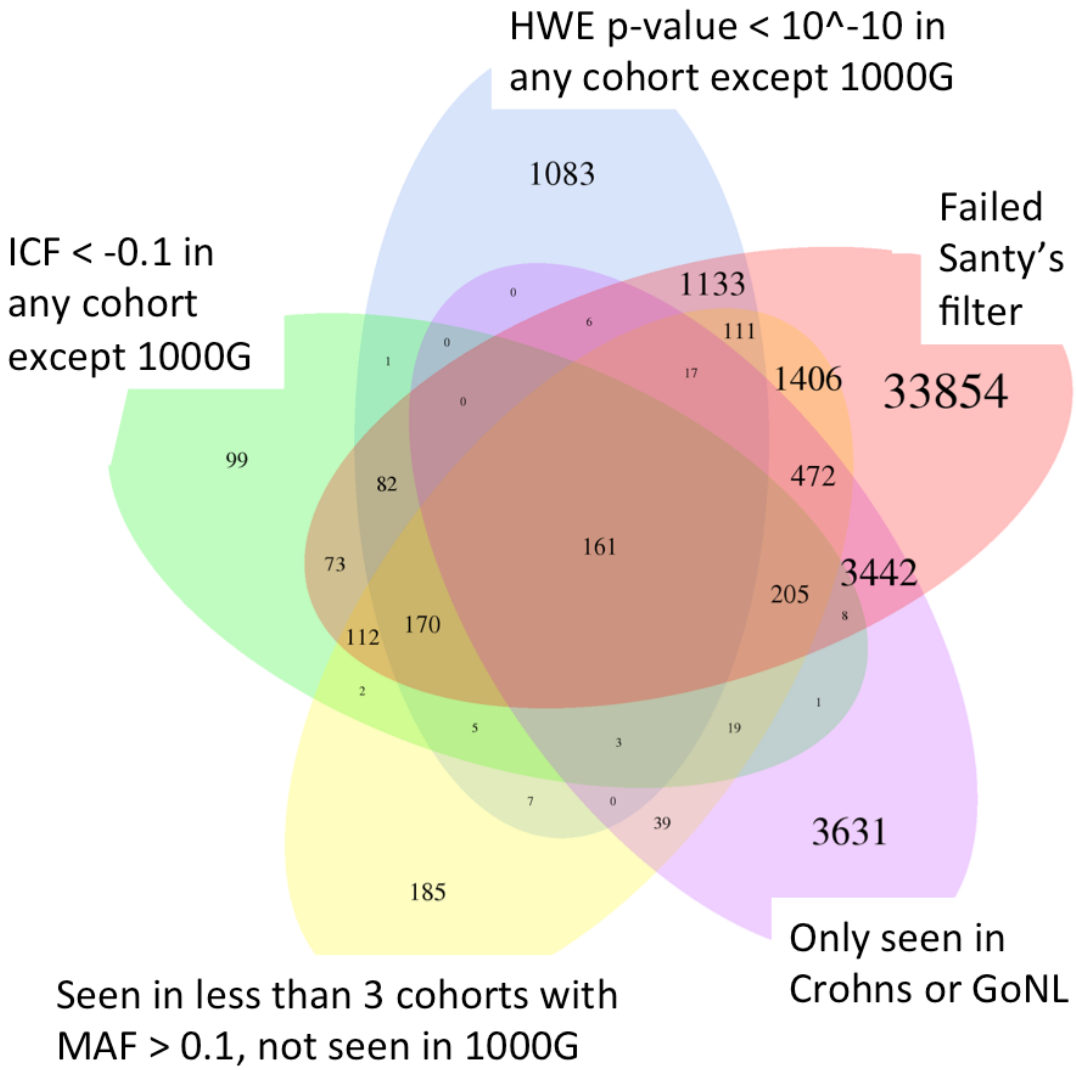

Figure 4.8: *Venn diagram of sites filtered out on chromosome 20.* The diagram depicts all cohort summary statistic-based filters and the cohort call rate-based filter marked as "Santy's filter". The cohort call rate filter captures the majority of filtered sites.

Another line of evidence of the quality of the final site filters is a comparison of per-sample Ts/Tv ratios performed by Shane McCarthy. The ratios look uneven across cohorts for the MAC5 site list (Figure 4.9, top). However, after application of

the final site filters, the ratios are similar between cohorts (Figure 4.9, bottom).

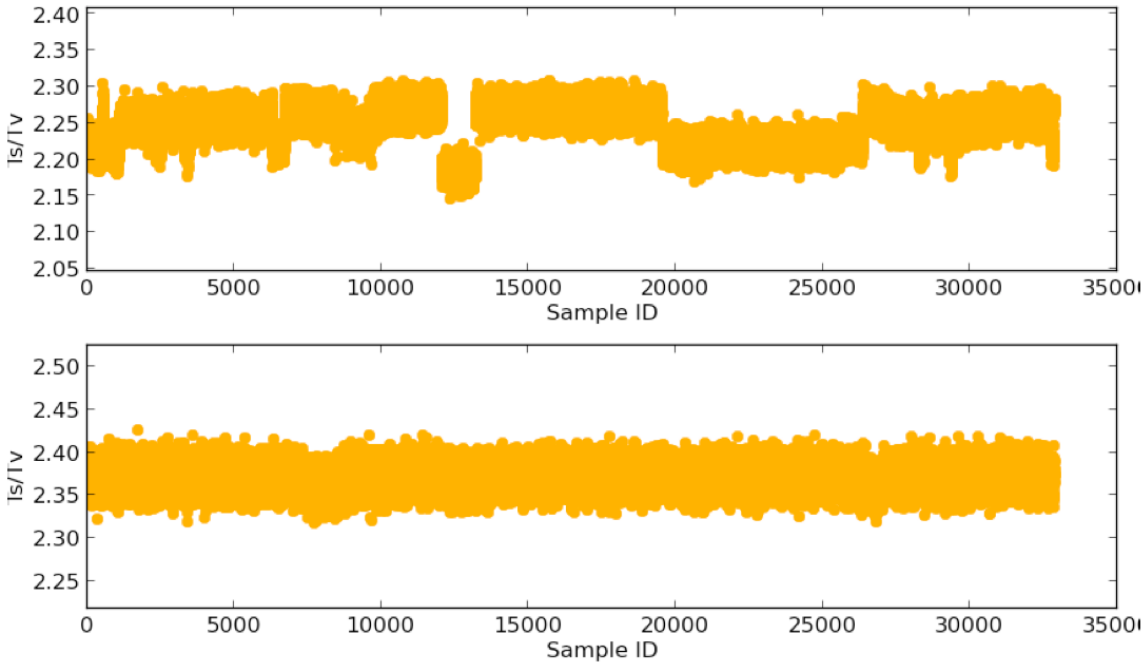

Figure 4.9: *Per-sample Ts/Tv ratio of HRC samples after applying the MAC5 filter (**Top**), and the MAC5 + final filters (**Bottom**).* This figure was prepared by Shane McCarthy. Samples are ordered by cohort. These samples include related samples. **Top:** Per-sample Ts/Tv ratios vary in chunks corresponding to cohorts. **Bottom:** The final site filters remove much of the cohort-specific variation in per-sample Ts/Tv ratios.

4.3.1.3 Downstream genotype imputation accuracy as a function of site filtering

Figure 4.10 demonstrates that filtering sites with our filters improves downstream genotype imputation accuracy of other sites on chromosome 20. However, the figure also shows that adding in multi-allelic SNPs to the site list degrades downstream genotype accuracy by a small amount. Table 4.3 shows that genotype imputation using our method improves when we apply our filters. Lastly, Figure 4.10 also shows that the haplotypes from GLPhase outperform the pre-existing haplotypes in downstream imputation accuracy.

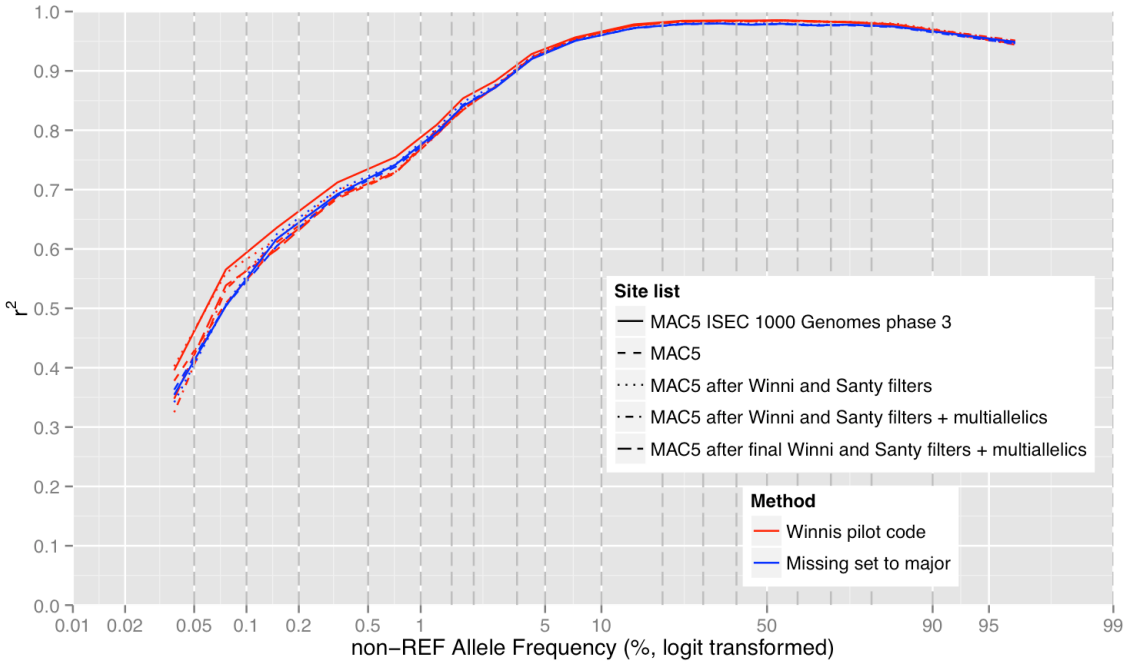

Figure 4.10: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of HRC panels created using different site lists of chromosome 20.* Genotypes imputed from haplotypes created using GLPhase are referred to as "Winnis Pilot code", genotypes imputed from the pre-existing haplotypes described in subsection 4.2.4 are referred to as "Missing set to major". These haplotypes were not rephased with SHAPEIT. Final HRC Santy filter haplotypes contain HRC release 1 samples with all related samples removed (32,611 samples). All other panels have only had the 1000 Genomes Phase 3 related samples removed (32,874 samples). Missing set to major haplotypes contain 32,905 samples. R^2 was calculated at 174,996 bi-allelic SNPs. Applying all filters improves genotyping accuracy at rare sites compared to all sites with a MAC < 5 ("MAC5") sites. Adding in multi-allelic SNPs reduces downstream genotype imputation accuracy by a small amount. Genotype discordances of all of these panels with an OMNI 2.5M chip are listed in Table 4.3.

Sites	Samples	Method	HR	Het	HA	NRD
HRC pilot AC10	Pilot	BEAGLE + SHAPEIT2	0.12	0.90	0.45	0.92
R1 MAC5	R1	MSTM	0.13	3.03	0.87	2.45
1000G Phase 3	R1 no TGP3R	WM + MSTM	0.06	0.33	0.24	0.39
R1 MAC5	R1 no TGP3R	WM + MSTM	0.06	0.35	0.23	0.39
R1 MAC5 with filters	R1 no TGP3R	WM + MSTM	0.05	0.33	0.22	0.38
R1 MAC5 with filters + multi-allelics	R1 no TGP3R	WM + MSTM	0.05	0.33	0.22	0.37
R1 MAC5 final filters + multi-allelics	R1 no TGP3R	WM + MSTM	0.05	0.32	0.23	0.37

Table 4.3: *Genotype discordance (in percent) between haplotype panels shown in Figure 4.10 and OMNI 2.5M genotypes available for 364 European samples from the 1000 Genomes project.* Discordances were measured at 42,229 SNPs on chromosome 20. The columns labeled "HR", "Het", "HA", and "NRD" contain the homozygous reference allele, heterozygous, homozygous alternative allele, and non-reference allele discordances, respectively. "R1" means HRC release 1. "no TGP3R" means 1000 Genomes Project Phase 3 without related individuals. "WM + MSTM" means GLPhase using missing set to major version three haplotypes as pre-existing haplotypes. The missing set to major panel has a high discordance rate compared to the other reference panels.

4.3.1.4 Addition of common SNP chip sites

Many future users of the HRC haplotypes will have SNP genotyping chip data and will expect all of the SNPs on their chip to exist in the HRC haplotypes so that all of their genotyped sites can be used for phasing. To accommodate this expectation, the HRC decided to create a site list of common SNP chips that would be added back into the site list after filtering. Sayantan Das performed the collation of SNP chip sites from twelve SNP chip arrays, and provided me with a site list. After applying the cohort summary statistic, and cohort call rate filters, we added back all sites in the SNP chip site list. Table 4.4 contains the counts for the final site list on chromosome 20, and across the autosomes. The SNP chip sites Sayantan sent me contained 4,914,335 sites. However, as can be seen from Table 4.4, only 148,570 of these sites were not already in the HRC site list.

Filter	Chr. 20	Autosomes	% change
MAC < 5	1,064,306	47,406,508	100.00
Cohort summary statistic-based filters	1,052,732	46,852,018	98.83
Santy filter	995,042	44,038,997	92.90
Add SNP chip sites	998,447	44,187,567	93.21

Table 4.4: *Incremental application of all filters to create final site list.* "Chr. 20" and "Autosomes" refer to the number of sites on chromosome 20 and the autosomes, respectively. All data are calculated after removing GPC site calls.

While creating the final site list, it came to my attention that a small number of lines in the VCF used to create Table 4.2 were duplicated in the VCF. This was due to a bug in the software written by Shane McCarthy to create the cohort summary statistics file. This bug was fixed before creating the final site list described in Table 4.4. Evidence of this coding error can be seen in the mismatch in counts between the "MAC < 5" row in Table 4.4 and the "MAC < 5, no GPC data" row in table 4.2 of the chromosome 20 column.

4.3.2 Downstream genotype imputation accuracy

One attribute of a method that scales well with sample size is that downstream genotype imputation accuracy increases with reference panel sample size. To show this, we ran GLPhase on successively larger sample subsets of the HRC, and then compared the resulting haplotypes through downstream imputation. Figure 4.11 shows that downstream imputation quality increases as the size of the reference panel phased increases. This evidence gave us confidence that GLPhase benefits from increased sample sizes.

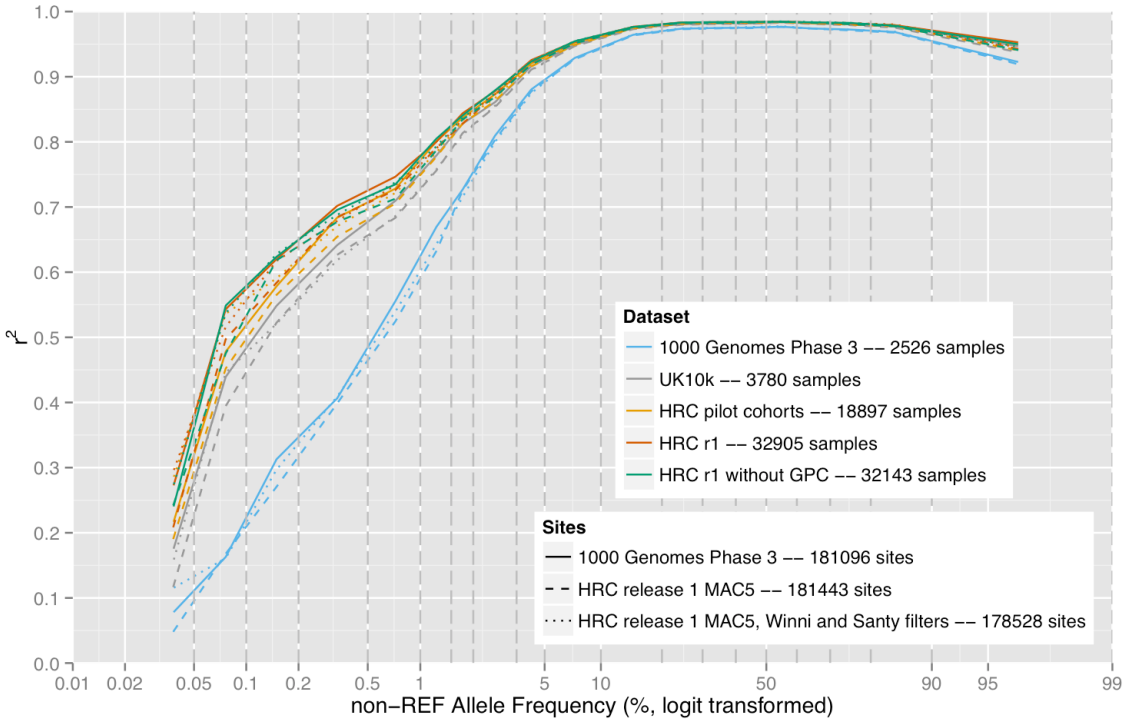

Figure 4.11: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of reference panels created using GLPhase from progressively larger sample sizes and at different site lists.* These haplotypes were not rephased with SHAPEIT. Downstream imputation quality increases with size of the sample set being phased. The largest gain in imputation accuracy is in rare variants.

The effect of rephasing the haplotypes created with GLPhase using SHAPEIT2 and SHAPEIT3 was also investigated. Figure 4.12 shows that rephasing with SHAPEIT improves haplotype quality as measured by downstream imputation accuracy, and

that SHAPEIT3 does not perform worse than SHAPEIT2. The run time of SHAPEIT2 was approximately four times the run time of SHAPEIT3.

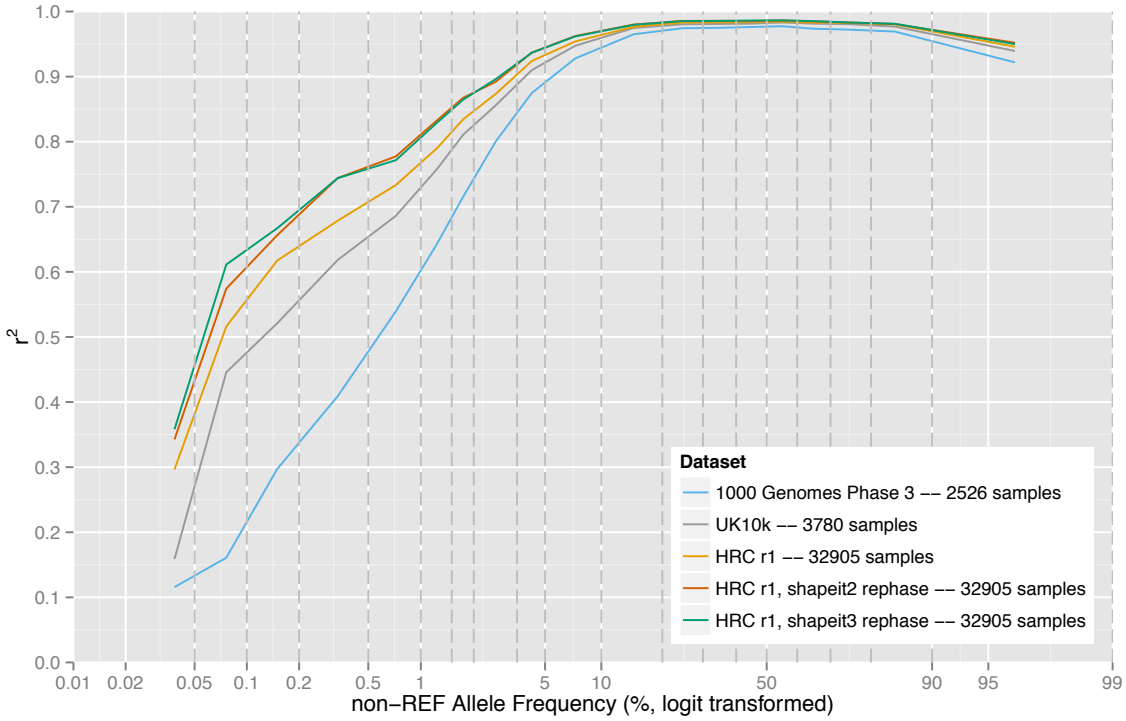

Figure 4.12: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of reference panels created using our method from progressively larger sample sizes, and rephased using SHAPEIT.* The MAC5 Winni and Santy filtered site list was used. Rephasing using SHAPEIT provides a boost in downstream imputation accuracy. SHAPEIT3 does not underperform in comparison to SHAPEIT2.

The quality of genotype imputation when using the HRC, UK10K, or 1000 Genomes Phase 3 haplotypes as a reference is compared in Figure 4.13. Using just GLPhase ("HRC release 1, no SHAPEIT" in Figure 4.13), we produce haplotypes of similar quality to the UK10K haplotypes. Rephasing with SHAPEIT provides a boost in imputation accuracy at rare sites.

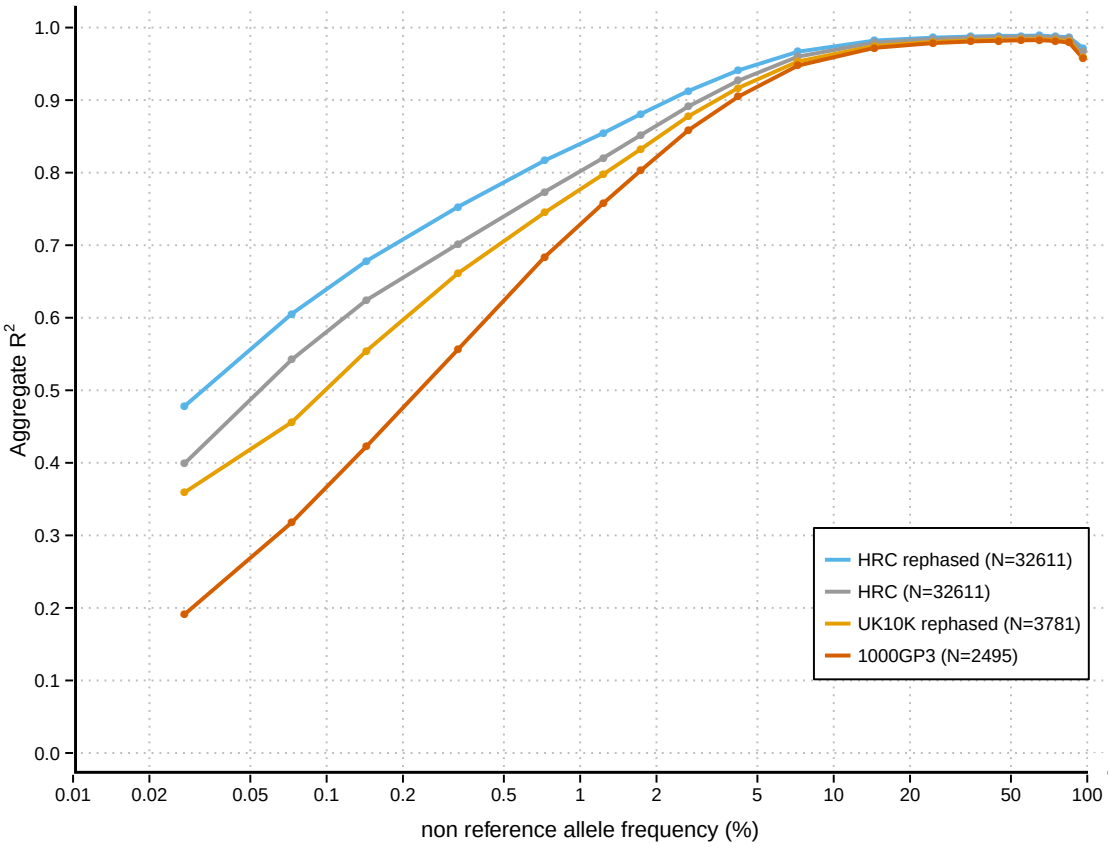

Figure 4.13: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of currently available European whole genome reference panels.* Comparison was performed at 7,750,151 autosomal SNPs that are present in all haplotype sets, and called in the 10 CG samples.

4.3.2.1 Effect of Impute2 conditioning haplotypes on downstream imputation accuracy

In their paper on genotype imputation strategies for reference panels with thousands of genomes, B. Howie, Marchini, et al. (2011) examine the effect of the number of conditioning haplotypes used by Impute2 on genotype imputation accuracy. When imputing Europeans, they show evidence that adding more than 500 conditioning haplotypes provides little gain in imputation accuracy of SNPs with a MAF $< 5\%$. Those tests were performed using the second release of the HapMap3 haplotypes, which is a SNP chip-based panel with far fewer rare SNPs than the HRC panel. It is instructive to re-examine the choice of conditioning haplotypes for the HRC panel. As shown in Figure 4.14, increasing the number of conditioning haplotypes when using the HRC panel only provides small increases in imputation accuracy for SNPs with MAF $< 0.5\%$. However, reducing the number of conditioning haplotypes from 500 to 100 does not appear to impact imputation accuracy much. This observation supports the hypothesis that as reference panels grow in size, the more likely it is that an individual can be found in the panel that is closely related to the imputed individual in the region being phased.

4.4 Discussion

This chapter describes the work that the HRC analysis group performed to create a unified haplotype reference panel of 32,488 predominantly European samples at 44,058,601 SNPs (44,187,567 VCF records after breaking multi-allelic SNPs into bi-allelics) across the autosomes. The panel improves downstream genotype imputation accuracy at sites with a MAF below 5% compared to the UK10K reference panel, the next best reference panel of Europeans available (see Figure 4.13).

The advantages of the approach taken to create the HRC reference panel are as

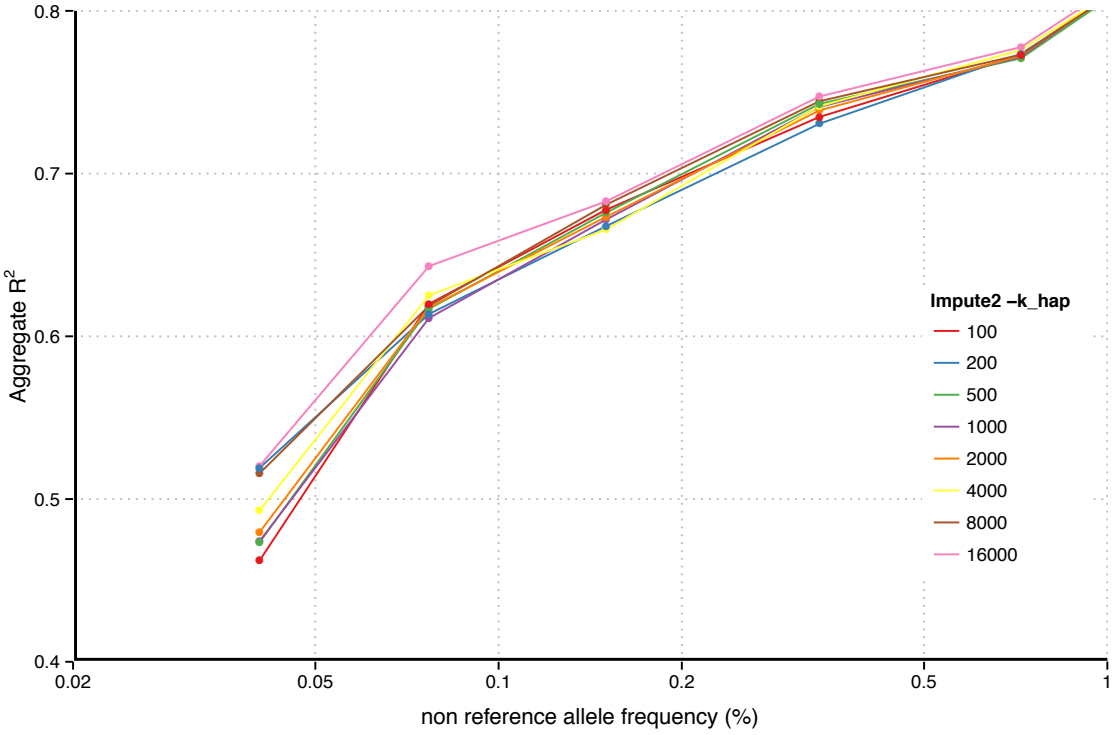

Figure 4.14: *Downstream genotype imputation accuracy as measured by squared Pearson correlation coefficient R^2 of final HRC panel using different numbers of conditioning haplotypes.* Only rare variants are shown, as other parts of the site frequency spectrum do not vary. Using 16,000 conditioning haplotypes outperforms using fewer haplotypes. Below and including 8,000 conditioning haplotypes, imputation accuracy does not appear to correlate with number of conditioning haplotypes used.

follows:

1. Due to the SNP filters, the quality of included SNPs is uniform across cohorts, as evidenced by per-sample Ts/Tv ratios before and after filtering (see Figure 4.9).

2. The panel genotype calls for known European 1000 Genomes samples are of a higher quality than the missing set to major and the HRC pilot genotypes (see Table 4.3).

3. The panel provides improved downstream genotype imputation accuracy for rare variants (MAF $< 5\%$) compared to imputation from UK10K (see Figure 4.12).

The disadvantages of the approach taken to create the HRC reference panel are as follows:

1. The improvement in downstream genotype imputation accuracy over the UK10K panel is modest compared to the order of magnitude difference in sample size between the panels (see Figure 4.12).

2. The approach requires access to aligned reads in order to create GLs for later merging into a single file. Aligned reads can take up a substantial amount of space. For example the ProjectMINE data were over 100 TB in size.

3. The approach requires access to pre-existing haplotypes for every sample.

4. The release does not include haplotypes for chromosome X. We are working on a haplotype release for chromosome X.

The next steps in the HRC project are to create chromosome X haplotypes, call indels, and increase the number of samples from non-European ethnicities. Creating chromosome X haplotypes is straight-forward, but has been delayed due to quality control problems with pre-existing haplotypes from some major projects. It is

expected that these problems will be fixed, and that work on chromosome X can continue.

Indel calling will need to be done by the HRC, as only few projects have submitted indel calls. Doing indel calling correctly will pose a technical challenge. Variant calling is most efficient if data from all samples in a genomic region can be compared together. This means that ideally all BAM files will need to be stored on the same system, instead of on two systems at the Sanger Centre and the University of Michigan as is currently the case. Once the data are on a single system, the reads need to be arranged by genomic coordinate instead of by sample. For the CONVERGE project, which only had 1.7x coverage data on 12k samples, this was a major difficulty, although there are shortcuts that can be taken to lessen the computational burden, such as reordering BAM files in batches of 500 samples and using lossy compression to further reduce the size of read data.

Including other ethnicities will double or triple the number of variants that will be part of the reference panel. Fortunately, no method investigated has a worse than linear time complexity in regards to number of variants processed. However, this added computational cost will need to be considered going forward.

Lastly, there are smaller tweaks to the genotype calling algorithm that could be explored further as described in section 3.7.

Chapter 5

Phasing Server

The genetic data of the Haplotype Reference Consortium (HRC) was contributed by 20 projects. Each of these projects used their own process of obtaining consent from the individuals they sequenced. Some of the projects did not obtain consent for their genetic data to be released to the public. As such, only a subset of the HRC haplotype panel will be made available to the public. Fortunately, all HRC informed consent agreements allow the use of the HRC haplotypes for phasing of arbitrary samples, as long as the HRC haplotypes remain on the University of Oxford network. In order to allow researchers to leverage the full size of the HRC haplotypes as a reference panel for phasing samples sequenced to high coverage, a server was written that phases uploaded genotypes without exposing the HRC haplotypes to the public. The server's home page is displayed in Figure 5.1 and can be found at `https://phasingserver.stats.ox.ac.uk`.

Two other servers exist that use the HRC haplotypes to impute genotypes into SNP chip data (see `https://imputation.sanger.ac.uk` and `https://imputationserver.sph.umich.edu`). However, this is the first web site designed to phase genotypes derived from high coverage sequencing data. Writing servers to allow scientists to perform analyses on data that cannot be easily shared is a growing tradition. Popular

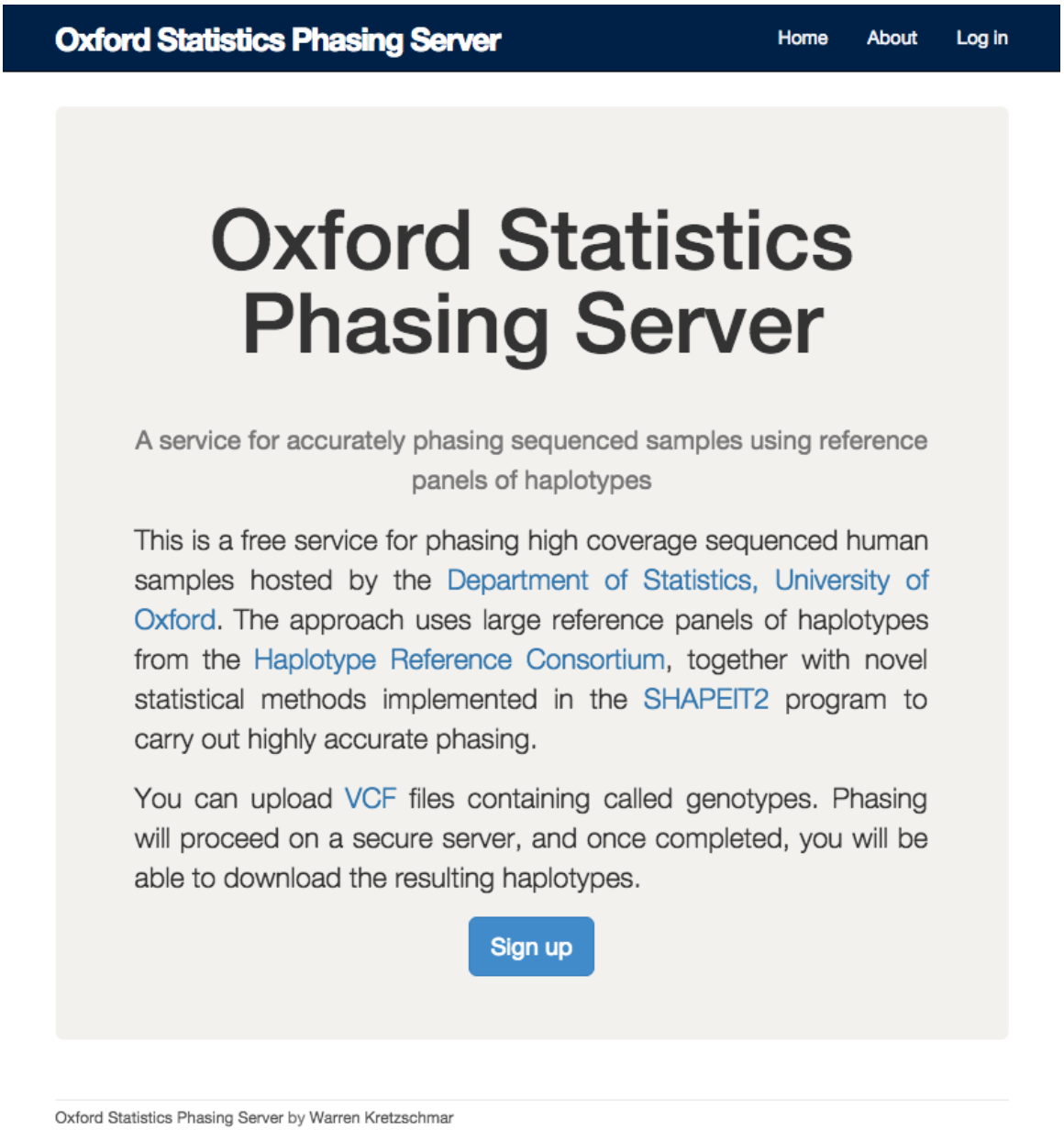

Figure 5.1: *Home page of the Oxford Statistics Phasing Server.*

examples of such web sites in bioinformatics are the UCSC Genome browser (Kent et al., 2002), the ENSEMBL project (Hubbard et al., 2002), and the bioinformatic analysis site GALAXY (Giardine, Riemer, and Hardison, 2005).

5.1 Application structure

The phasing server has a two part structure made up of a front end and a back end. Both the front and the back end are applications written in a programming language popular in the web design community called Ruby (`https://www.ruby-lang.org/`). The front end is written using a popular open source web framework called Ruby on Rails (`http://rubyonrails.org/`) and it provides the user interface. The back end is written in a simpler web framework called Sinatra (`http://www.sinatrarb.com/`) and is responsible for checking and phasing genotypes uploaded by users. The front end communicates with the back end through a narrow application program interface (API).

Only the back end has access to reference panels, phasing binaries, and phasing scripts. The front end stores user uploaded genotypes and related information, as well as meta-data on the reference panels and phasing scripts. Therefore, while the back end needs to be located inside the University of Oxford network, the front end can in principle be located anywhere on the web. Figure 5.2 is a pictorial representation of the phasing server structure.

5.1.1 Front end – the user interface

The front end is a Ruby On Rails application that handles all direct user interactions. Specifically, it provides web pages for handling the following server features:

- User registration, email validation, and password authentication
- Accepting VCF uploads from users

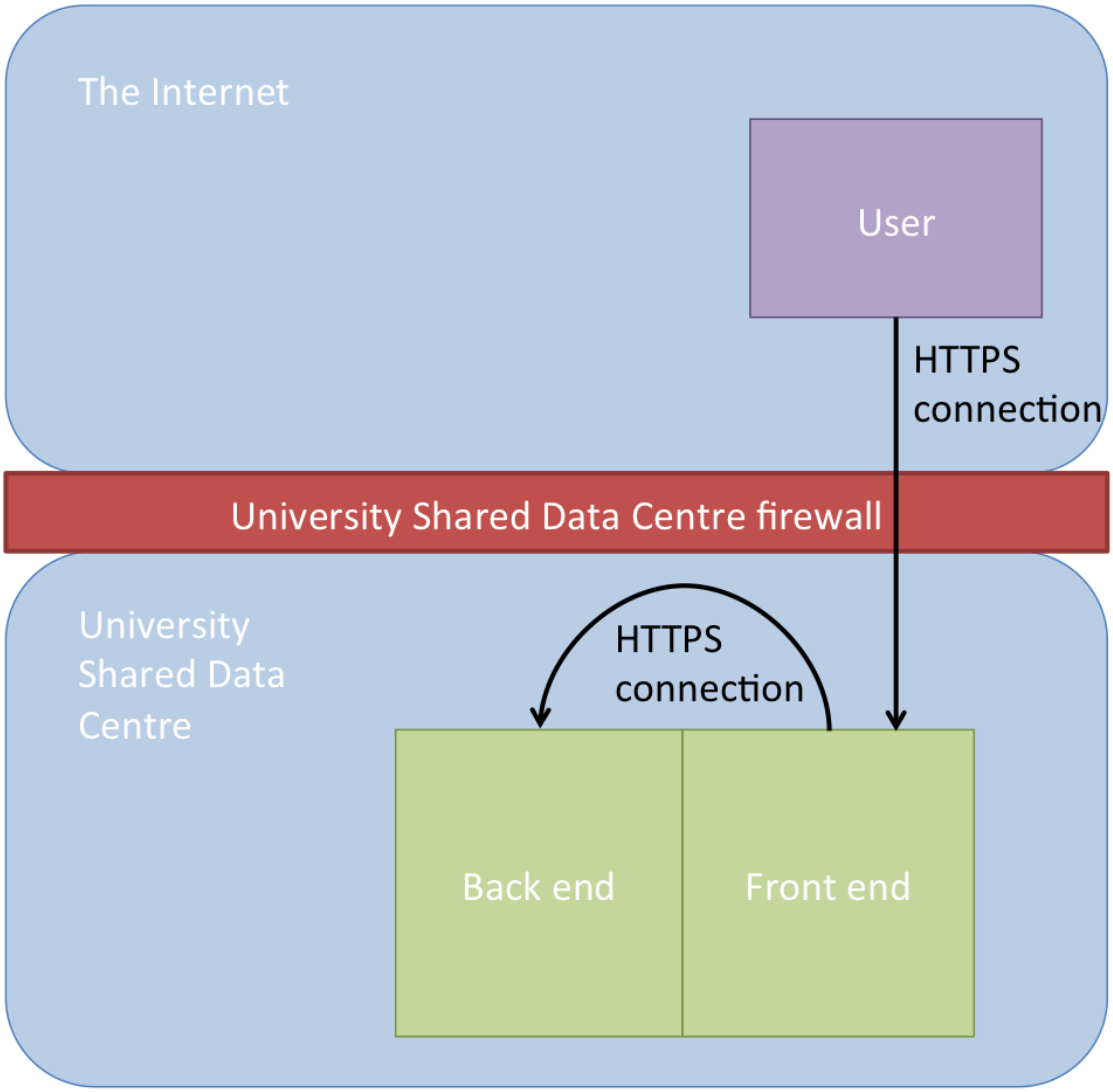

Figure 5.2: *Pictorial representation of the phasing server's network structure.*

- Accepting user requests to phase uploaded VCFs
- Providing users with results from checking and phasing uploaded VCFs
- Providing descriptions of the reference panels and phasing scripts available for phasing.
- Allowing users to download phased genotypes

5.1.2 Back end – the Application Program Interface (API)

The back end is a Ruby-based Sinatra application. It exposes a narrow API that can be used to control it. The back end handles the following server features:

- Storage of reference panels, binaries, and scripts for phasing
- Checking of VCF validity and alignment with reference panels
- Phasing of VCFs using stored reference panels

5.2 Security

Security is a primary concern for this application, as both the HRC haplotypes and the user uploaded genotypes are sensitive human genetic data. There are several measures in place that serve to provide security for the sensitive data stored by the application. The server will be located inside the University of Oxford network, which means that measures also need to be taken to secure the university network from attacks through the phasing server.

5.2.1 Physical security

These security measures are designed to restrict unauthorized access to the hardware running the application. Without physical security there is no data security

(Subashini and Kavitha, 2011).

Both the back and front ends are located on a dedicated server behind a firewall at the University of Oxford's shared data center (USDC). The USDC has been designed with security in mind and has several industry standard measures in place to ensure the server is physically secure. These include:

- a biometric, proximity reader entrance system
- an anti-tailgating security portal
- a proximity card access to racks
- CCTV systems
- a room access audit trail of who entered and exited at what times
- an audit trail of who opened which rack and for how long

5.2.2 Attack surface minimization

These security measures are designed to reduce the number of attack vectors of a system. An attack vector is a way by which an attacker can gain unauthorized access to a system. The fewer attack vectors a system has, the smaller the chance is that an attacker could exploit any one of them to gain unauthorized access.

The phasing server only exposes two ports to the Internet: The standard HTTPS server port and a secure shell port. The secure shell port only accepts connections originating from IP addresses inside the statistics department.

5.2.3 Operating system maintenance

Like any piece of software, the server operating system has unknown bugs that may be discovered at a future date. Some of these bugs may allow the server's security

to be compromised[1]. This is why a server that is exposed to the Internet needs to be regularly updated to remain secure. The phasing server's operating system is maintained and kept up to date by Statistics Department IT personnel.

5.2.4 Vulnerability mitigation

No security system is perfect. Therefore, it is important to minimize the damage that can be done by successful exploitation of an attack vector.

Even though the back and front ends are located on the same server, they run as two separate low-privilege users, which means that they do not have access to each other's data. As the back end is only accessible from applications running on the same physical server, this serves as a mitigation strategy against a successful attack on the front end. If the front end is compromised by an attacker, that attacker can only communicate to the back end through its API, but does not gain direct access to the HRC haplotypes. However, one could imagine a situation in which an attacker could leverage the API to make inferences about the HRC haplotypes: If an attacker wants to check if a particular pair of haplotypes is in the HRC, the attacker could submit their haplotypes through the API. The switch error of the rephased haplotypes returned by the API could be used to gauge if someone closely related to the input sample exists in the HRC.

To minimize the impact of a successful attack on the phasing server, all user uploaded genotypes are only retained for a week after they are last used by the user to create phased haplotypes. All phased haplotypes are also only retained for one week.

The phasing server is isolated from the rest of the university network so that even if an attacker compromises the phasing server completely, the attacker will not be able to gain any further access to the rest of the university network.

[1] see for example the Heartbleed bug (Durumeric et al., 2014).

5.2.5 Connection encryption and authentication

Connection encryption and authentication safeguards against attackers eavesdropping on communications. It is important to note that authentication is necessary in order to guarantee that the communication is being encrypted for the intended recipient and not a third party posing as the intended recipient.

All communications between the front end, back end, and user are encrypted through the industry standard Hypertext Transfer Protocol (HTTP) (Fielding et al., 1999) over Transport Layer Security (HTTPS) protocol (Dierks and Rescorla, 2008). HTTPS provides both encryption of messages sent over an insecure network, and authentication of sender and receiver of those messages.

5.3 Access

The phasing server will be made available for use by researchers and clinicians who have small numbers of whole genome sequenced samples that they wish to phase. To gain access, a user will need to send the server administrator an email asking to be put on an email white list. The email white list is accessible to server administrators through the email white list page shown in Figure 5.3. The interface has controls for:

- adding emails
- authorizing emails for a duration of time
- setting the maximum number of samples allowed per uploaded VCF
- deleting emails

Only users with authorized emails on the white list can create accounts, upload data, and start phasing jobs.

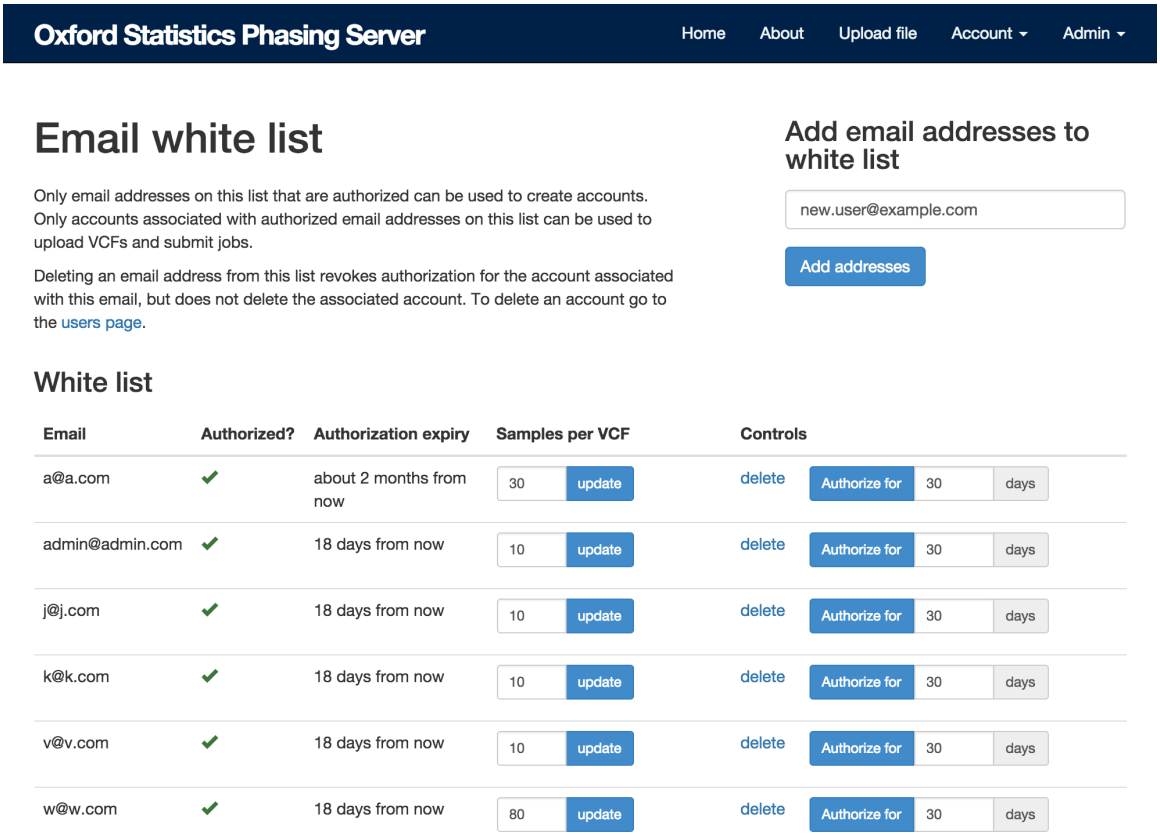

Figure 5.3: *Administrator interface for adding user emails to the email white list.* The authorization duration and cap on samples in an uploaded VCF can be changed here as well.

5.4 Data preparation

The phasing server does not do any pre-processing of genotypes uploaded by the user. Instead, the server has a detailed description of how genotype data needs to be processed before it can be uploaded to the server. The server checks to make sure the uploaded genotypes conform to expectations, and propagates an error message back to the user if the genotypes are malformed.

The server expects the uploaded genotypes file to:

1. be a valid VCF
2. be a block compressed gzip file
3. be indexable using Bcftools
4. only contain bi-allelic SNPs or Indels
5. only contain genotypes
6. not contain missing genotypes
7. only contain variants that have the same coordinate, REF, and ALT allele as a variant in the reference panel site list
8. contain chromosomes that have the same name as the chromosomes found in the reference panel site list

Below is explained how to make sure the data conforms to all of these requirements. Similar instructions can be found on the phasing server.

5.4.1 Preparing the genotypes using Bcftools

Bcftools is an open source tool for working with VCF and BCF files that is maintained primarily by researchers at the Sanger Institute. The instructions for preparing the

genotype data use this tool because it is fast, and flexible enough to perform all the necessary operations. Bcftools depends on htslib. Instructions for installing Bcftools can be found at `http://www.htslib.org`.

A script that "just works" can be found in appendix C[2]. To use the script, the VCF needs to be called `geno.vcf.gz` and the site list of the target reference panel needs to be called `var_list.vcf.gz`[3]. Those two files need to be copied into a new directory together with `check_vcf.sh`. This command can be run to check the VCF:

```
bash ./check_vcf.sh
```

If all checks pass, then the last line of the output should read:

```
ALL CHECKS OK
```

5.4.2 Breaking down the VCF checks

For all examples we will assume a Bash environment on a UNIX-like operating system that has Bcftools installed. Furthermore, we will assume that the genotypes file is stored under the current directory and has the name `geno.vcf.gz`. For some examples a site list is required, and we will assume that it has been downloaded to the current directory and renamed to `var_list.vcf.gz`.

It is recommended to work through these examples in order, as later examples will make assumptions about the input file that are checked by earlier examples.

5.4.3 Checking the VCF is valid

The first thing to try is to parse the VCF using Bcftools and see if it throws any errors:

[2]The script can also be downloaded from `https://phasingserver.stats.ox.ac.uk/static_pages/check_vcf.sh`.

[3]The site list for the HRC can be downloaded from `https://phasingserver.stats.ox.ac.uk/site_lists/hrc_r1_20150417/HRC.r1.GRCh37.autosomes.mac5.sites.vcf.gz`.

```
bcftools view geno.vcf.gz >/dev/null
```

This reads the VCF and redirects the standard output to `/dev/null`. If there is any output, then the output must be coming from `STDERR`, and therefore is indicative of a parsing error.

5.4.4 Block compressing the VCF

Bcftools can be used to block compress the VCF and store the block compressed version in `out.vcf.gz`:

```
bcftools view geno.vcf.gz -Oz -o out.vcf.gz
mv out.vcf.gz geno.vcf.gz
```

5.4.5 Making sure the VCF is indexable by Bcftools

Indexing the VCF allows Bcftools to quickly extract chromosomes and regions from the VCF. Only block compressed VCFs can be indexed. The following command indexes the VCF and should not throw any errors:

```
bcftools index geno.vcf.gz
```

5.4.6 Keeping only biallelic SNPs and Indels

This command only keeps variants that contain exactly two alleles (one REF and one ALT allele), and are either SNPs or Indels:

```
bcftools view -m2 -M2 -v snps,indels geno.vcf.gz -Oz -o out.vcf.gz
mv out.vcf.gz geno.vcf.gz
```

5.4.7 Making sure the VCF contains only genotypes

The VCF should only contain the `GT FORMAT` tag. This command drops all other `FORMAT` tags:

```
bcftools annotate -x FORMAT geno.vcf.gz -Oz -o out.vcf.gz
mv out.vcf.gz geno.vcf.gz
```

5.4.8 Making sure the VCF does not contain any missing genotypes

The command below checks every line of the VCF to look for any missing genotypes (something like `./.` or `.`). All lines that have at least one missing genotype are removed. Before this step one might first want to remove individuals with high rates of missing genotypes, but that is beyond the scope of these examples.

```
bcftools view -g ^miss geno.vcf.gz -Oz -o out.vcf.gz
mv out.vcf.gz geno.vcf.gz
```

5.4.9 Making sure the VCF contains the chromosomes in the variant list

```
bcftools index geno.vcf.gz
for i in $(seq 1 22); do
  bcftools view -G -r $i geno.vcf.gz >/dev/null
done
```

If the code above throws any errors regarding chromosomes, then the chromosomes in the VCF might need to be renamed. This can be done by creating a map from the current chromosome names to the numbers 1 through 22 and then using the command `bcftools annotate --rename-chrs`.

5.4.10 Subsetting the VCF down to the reference panel site list

First we index both the genotypes and site list files and then use Bcftools to intersect the files.

```
bcftools index geno.vcf.gz
bcftools index var_list.vcf.gz
bcftools isec -n=2 -w1 geno.vcf.gz var_list.vcf.gz -Oz -o out.vcf.gz
mv out.vcf.gz geno.vcf.gz
```

5.5 How to use the server

This section is a short tutorial of the phasing server.

5.5.1 Creating an account

To gain access to the phasing server, a user must first create an account. To create an account, the user emails the phasing server administrator asking for access. The administrator adds the email address to the phasing server's email white list as shown in Figure 5.3. The user's email address is automatically authorized to create an account, upload VCFs, and create phasing jobs for those VCFs for one month. The user then creates an account on the sign up page shown in Figure 5.4. An account activation email is sent to the user's account with a link to activate the user's account. Once the user clicks on that link, the account is activated and the user can log in.

5.5.2 Logging in

To log in, the user needs to click the "Log in" link at the top right of the home page shown in Figure 5.5. On the following page (shown in Figure 5.6), the user will need to type in their email address and password.

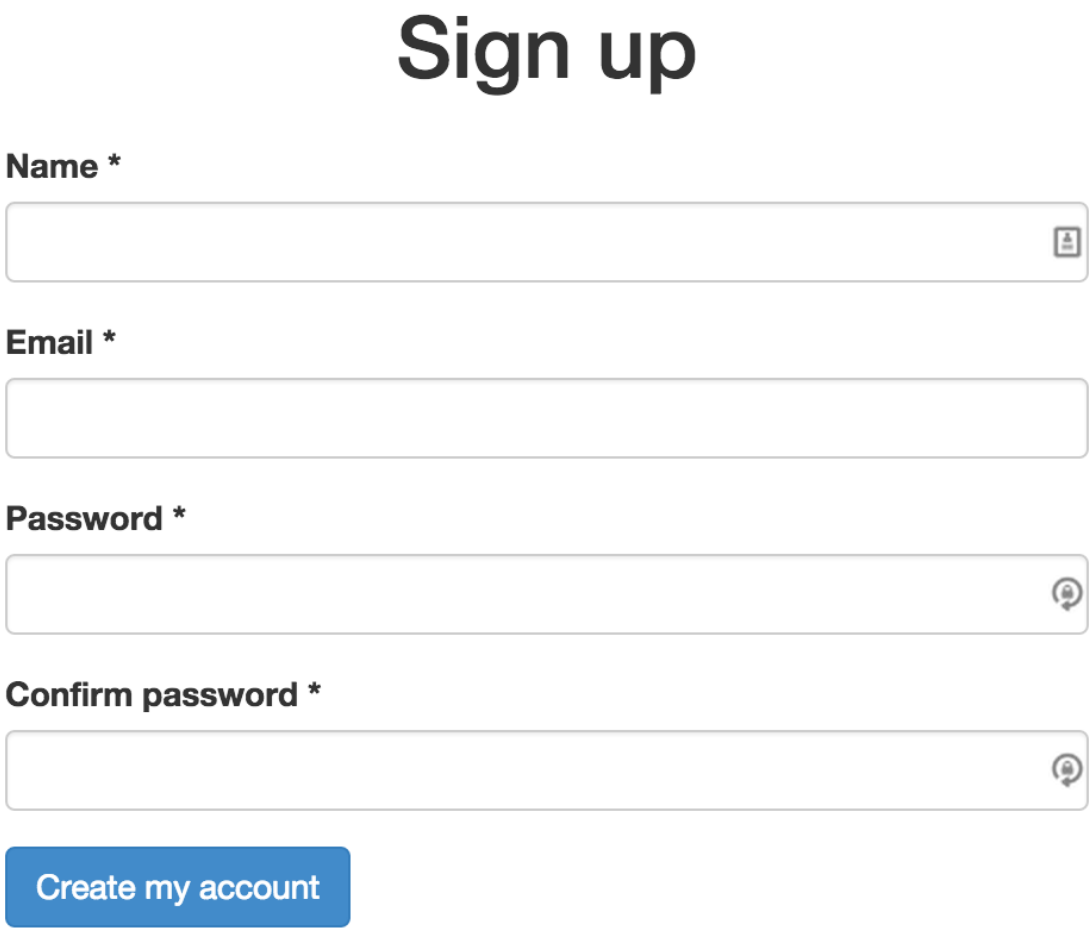

Figure 5.4: *Form for creating a user account.*

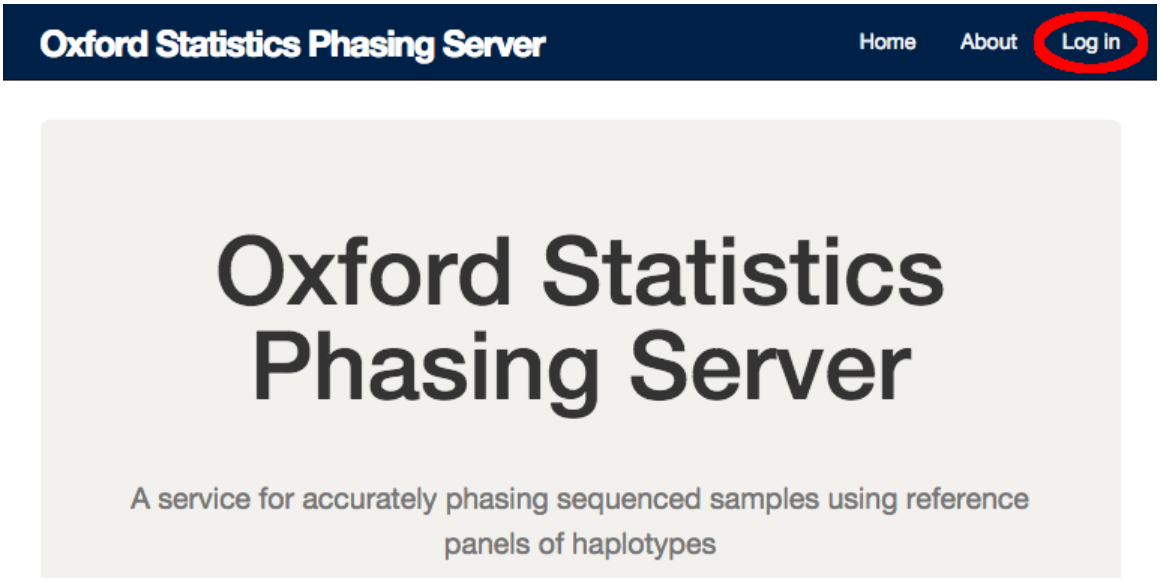

Figure 5.5: *Location of the log in link on the home page.*

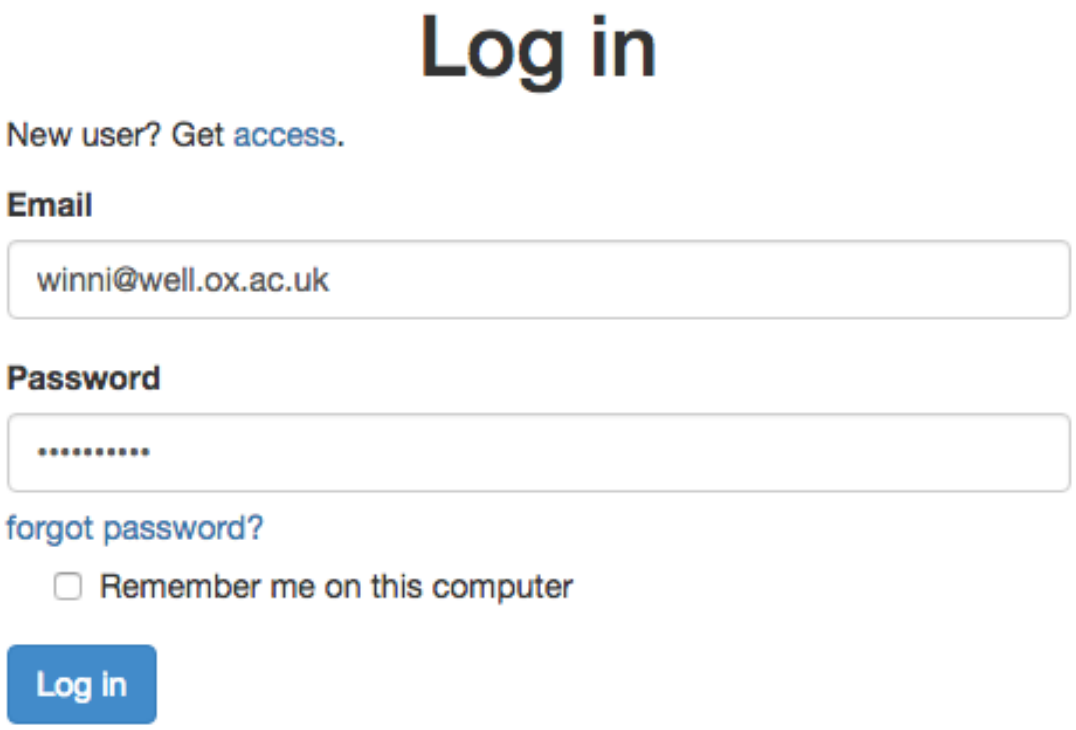

Figure 5.6: *Example of log in page.*

5.5.3 Uploading a VCF

After logging in, a link called "Upload file" appears in the navigation panel at the top right of the page. Clicking on it opens up the VCF upload page as shown in Figure 5.7.

On the upload page the user specifies a project name for the VCF, a reference panel to use for phasing, which chromosomes are contained in the VCF, and the location on the computer where the VCF is located. Multiple chromosomes can be selected using the shift key. Optionally, a Phase Informative Reads (PIR) file can be uploaded as well in order to allow SHAPEIT2 (Delaneau, Zagury, and Marchini, 2013) to use phase informative reads (Delaneau, B. Howie, et al., 2013). The SHAPEIT website[4] makes a tool available for creating PIR files from aligned reads in BAM format. However, if a PIR file is used, then the VCF may only contain one chromosome. More details

[4] `http://www.shapeit.fr`

Figure 5.7: *Form for uploading genotype VCFs.*

Figure 5.8: *Panel representing a recently uploaded VCF.* The label in the top right indicates the status of checking the VCF for errors. The "view" button on the bottom left can be used to navigate to a page containing more information about the VCF. The "delete" button on the bottom right can be used to delete the VCF.

on the phasing program used can be found in section 5.6.

After clicking the "Upload file" button, the user is redirected to their home page that now displays a panel representing the uploaded VCF as shown in Figure 5.8. The VCF is automatically checked for errors. A label in the top right indicates the status of checking the VCF for errors. Buttons on the bottom left and right allow navigation to a page with more information about the VCF, and deletion of the VCF, respectively.

5.5.4 Phasing

Once the VCF passes all checks, a panel appears underneath the VCF panel that allows the creation of a job to phase one of the chromosomes in the VCF as shown in Figure 5.9. Clicking the "Phase Chromosome" button starts a new phasing job. After the phasing job has started, the status of the job will automatically update itself as shown in Figure 5.10.

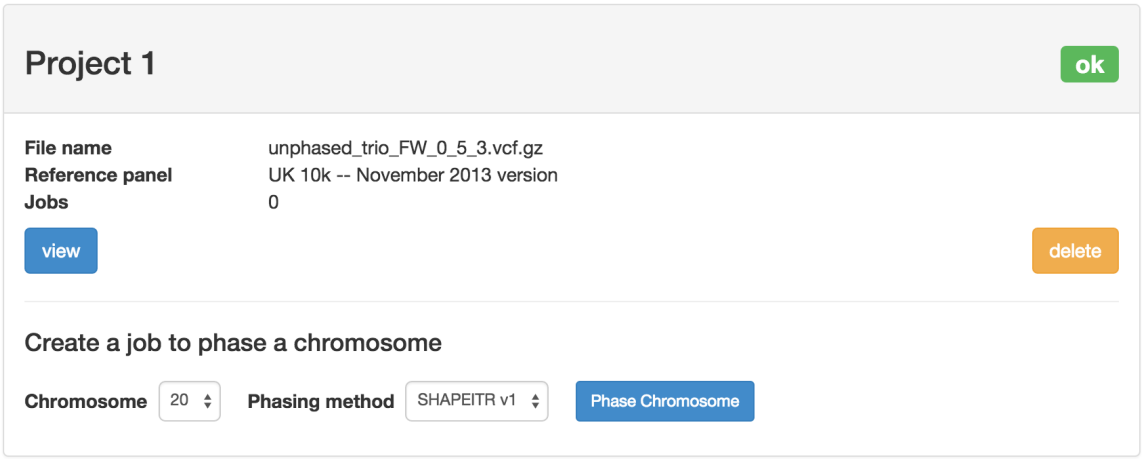

Figure 5.9: *Panel representing an uploaded VCF that has passed all checks.*

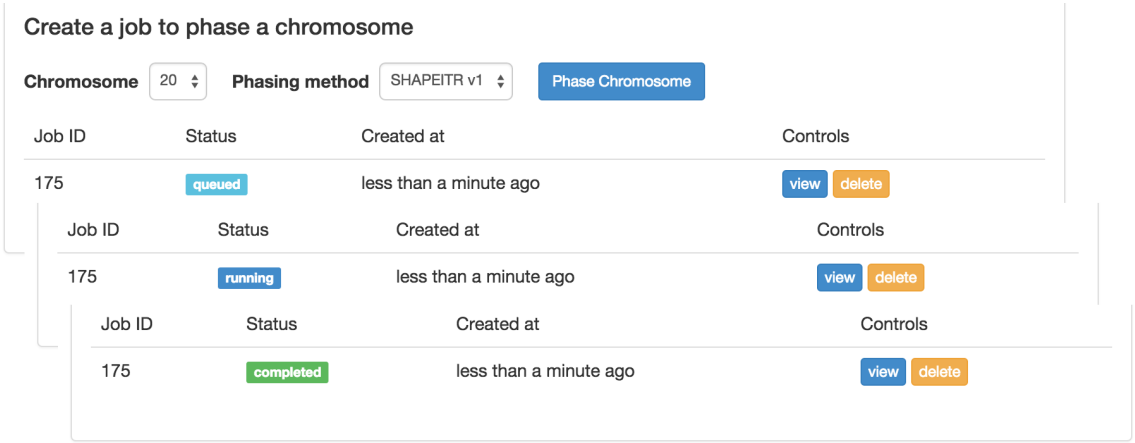

Figure 5.10: *Compilation of a series of job status panels showing how the job status is updated over time.*

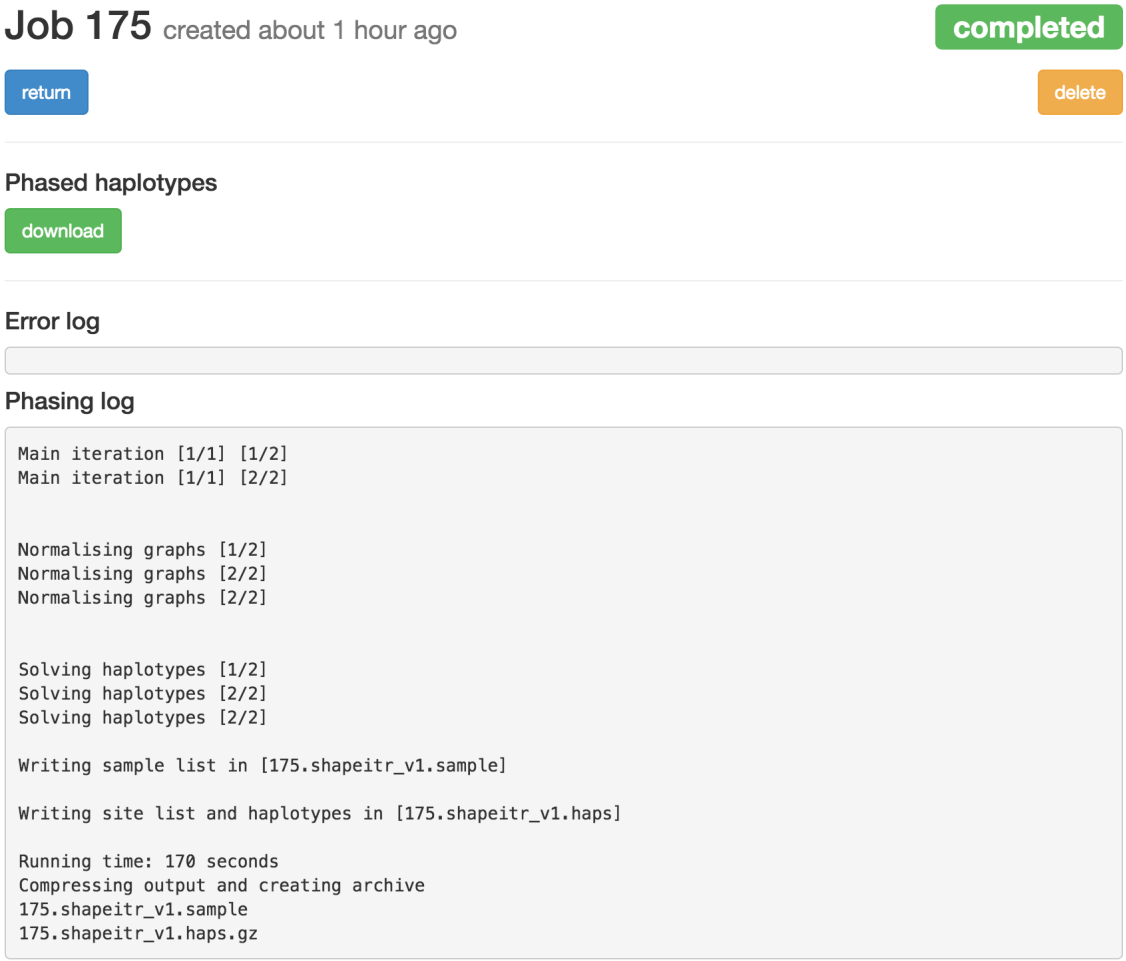

Figure 5.11: *Jobs page with button to download phased genotypes.* The phasing log contains the system output from running the phasing software.

5.5.5 Downloading phased haplotypes

Once the status of the job is "completed", it can be downloaded by first clicking on the "view" button and then the "download" button on the job page shown in Figure 5.11.

5.5.6 Choosing a reference panel

Currently, the server supports two reference panels: The April 2015 release of the Haplotype Reference Consortium and the November 2013 version of the UK10K panel (UK10K Consortium, 2015). Figure 5.12 shows how information relating to the panels are displayed on the server. Both panels contain about 40 million bi-allelic SNPs, but

the UK10K panel contains some bi-allelic indels, whereas the HRC panel contains some tri-allelic and quad-allelic SNPs. The downloads section contains a link to the site list of each reference panel. The SHA-256 sum can be used to verify the integrity of the downloaded VCF.

5.6 Choice of phasing program

Although the web server can in principle use any phasing method, the method used needs to be relatively fast in order for the server to be able to return the phased haplotypes to the user quickly. Currently, the server uses a modified version of SHAPEIT2 (Delaneau, Zagury, and Marchini, 2013) called SHAPEITR (Sharp et al., 2016). SHAPEITR exploits the abundance of rare variants in the UK10K and HRC panels to find a good set of neighbors for SHAPEIT2 to phase the supplied genotypes with. The SHAPEITR method for finding nearest neighbors across a chromosome is as follows:

1. The chromosome is split into equally sized regions starting from a randomly chosen start location.

2. For each window, starting with a minor allele count of one, haplotypes of individuals are added to a list of haplotypes that share alleles of the given allele count with the sample to be phased. The minor allele count of sites considered is increased until a pre-defined number (K) of haplotypes have been added to the list.

3. In each window, the haplotypes on the window's list are used as the nearest neighbors for the SHAPEIT2 algorithm while it phases the sample. Phasing in this way is performed only once, and no MCMC iterations are run.

Sharp et al. (2016) show that this method of choosing nearest neighbors is an

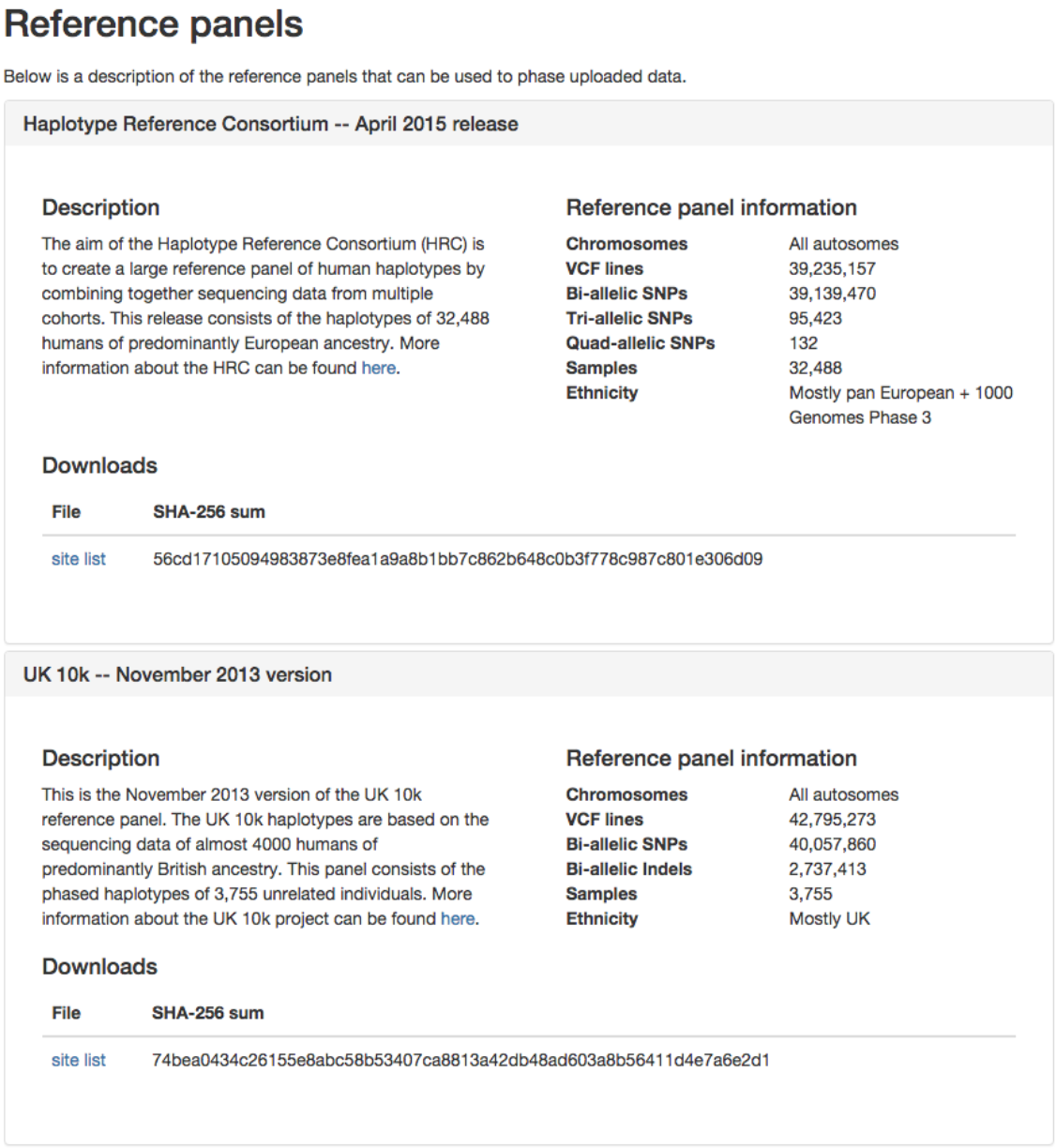

Figure 5.12: *Description of available reference panels as it appears on the phasing server.*

order of magnitude faster and leads to better haplotypes (as measured by switch error) than using SHAPEIT2 alone, when phasing a high coverage sample (about 130x Illumina sequencing) from the UK10K reference panel (UK10K Consortium, 2015). A requirement for SHAPEITR to work is that the unphased genotypes have many rare variants. This is why the unphased genotypes should be of high density such as would for example be the result of high coverage sequencing.

Phase informative reads (PIRs) are sequencing reads that overlap at least two sites at which the sequenced sample is heterozygous. The SHAPEIT2 model uses PIRs to constrain its model such that the model is more likely to sample haplotypes consistent with PIRs. This increases haplotype quality as measured by switch error (Delaneau, B. Howie, et al., 2013). Both SHAPEITR and SHAPEIT2 can use PIRs, and the phasing server passes uploaded PIR files on to these phasing programs.

5.7 Discussion

The web server described in this chapter provides an interface that allows researchers that do not have access to the HRC haplotypes to use the HRC panel to phase their own samples. The server ensures that only authorized users have access to the server. Once an authorized user has authenticated themselves to the server, the server accepts genotypes in VCF format, which it checks to make sure they conform to a list of quality standards. If the VCF passes these checks, then it is phased from the UK10K or the HRC reference panel using SHAPEITR, which exploits rare variants in the reference panel to phase the samples quickly and accurately. After phasing is complete, the user can download the phased haplotypes. All transactions between the user and the server are encrypted. The phasing server described in this chapter will be published as part of Sharp et al. (2016).

Although there is a server at the Sanger Centre (`https://imputation.sanger.`

ac.uk) and a server at the University of Michigan (https://imputationserver.sph.umich.edu) that allow genotype imputation into SNP chip data from the HRC panel, neither of these servers offers a service for phasing genotypes derived from whole genome sequencing. It may be the case in the future that either of these servers offer a phasing service for whole genome sequenced data, but currently the focus of these services appears to be the imputation of genotypes into large SNP chip-based GWAS panels.

Another problem with the other services is that they are public. SHAPEIT is only free for academic use, which poses obstacles to introduction of SHAPEIT on the Sanger and University of Michigan servers. This server side steps that problem by only being available for academic use.

Chapter 6

Conclusion

This thesis explores methods for genotype calling and phasing from low coverage sequencing data.

As part of this exploration, chapter 2 describes a pipeline to use current methods to genotype and phase around 12,000 CONVERGE samples sequenced to sparse coverage (1.7x). CONVERGE Consortium (2015) use these genotypes in a GWAS to discover the first replicated novel associations with major depressive disorder.

Chapter 3 describes how the SNPTools model was developed into a method for genotype calling and phasing of GLs that scales better with sample size than competitor methods and makes the unified calling and phasing of genotypes from GLs in tens of thousands of samples possible.

Chapter 4 showcases how my model was used to create a unified haplotype call set from the HRC data set, the largest collection of whole genome sequencing data in the world. The HRC haplotypes have already been used to improve the resolution of genetic associations in some of the HRC member cohorts (McCarthy et al., 2015). The description of my model and the HRC analysis is available as a pre-print[1] manuscript that is currently under review (McCarthy et al., 2015).

In order to make the HRC panel available as a reference panel to the academic

[1] http://biorxiv.org/content/early/2015/12/23/035170

community, chapter 5 describes a server that phases submitted high-density genotypes from the HRC panel. This work is under review as part of Sharp et al. (2016).

The methods presented here allow us to extend the sample size at which it is feasible to call and phase genotypes from GLs by an order of magnitude. With changes to the clustering implementation of GLPhase it should be possible to extend the feasible sample size even further.

From the perspective of GWAS, low coverage sequencing at around 4x coverage has been a successful method for discovering new variation in the human genome and making it available to existing SNP chip-based GWAS through imputation from haplotype reference panels. Building on the pioneering work of the 1000 Genomes Project, the projects that contribute to the HRC have allowed the expansion of the size of European reference panels, which has lead to an increase in the accuracy of genotype imputation into SNP chip GWAS samples. The HRC plans to expand the populations included in the HRC in subsequent releases, and it is likely that more mileage is to be gained from increasing the size of low coverage sequencing-based reference panels for genotype imputation into SNP chip-based GWAS.

However, the cost of high coverage sequencing has been coming down rapidly, and today it is possible to sequence a human genome to high coverage for less than 1,000£. In the next few years it is expected that tens of thousands of high coverage genomes will become available for medical genomics through the 100,000 Genomes Project[2]. If the cost of sequencing continues to fall as it has in the past ten years, and if massive medical sequencing projects like the 100,000 Genomes Project are successful, then it may well be that due to supply, future haplotype reference panels will consist of many high coverage genomes as well.

The fate of sparse genome sequencing (less than 2x coverage) is less certain. On the one hand, this thesis validates ideas of Pasaniuc et al. (2012) and others by showing

[2] `http://www.genomicsengland.co.uk`

the utility of sparse sequencing as a replacement of SNP chips in the CONVERGE GWAS. On the other hand, chapter 2 also makes clear that sparse sequencing is substantially less well suited than 4x coverage sequencing for the creation of haplotype reference panels. Furthermore, there are hurdles to introducing sparse sequencing as a replacement for SNP chip-based GWAS: Sparse sequencing is still several times more expensive than SNP chips (see chapter 2), and it has yet to be established that genotype imputation accuracy of sparse sequencing grows with reference panel size at the same rate that imputation into SNP chips does.

The rise of large medical genomics data sets such as the 100,000 Genomes Project also highlights the utility of servers such as the one presented in chapter 5. The 1000 Genomes Project did not measure phenotypes in order to allow all 1000 Genomes Project data to be made public. With medical genomics data sets, sharing of sequencing data in the same way will likely be impossible. However, once these data sets exist, web servers will play an important role in allowing researchers to phase and impute genotypes from these valuable data sets without having to be trusted with sensitive information.

Appendices

Appendix A

Detailed Instructions for calling GLs in the HRC

Detailed instructions for calling GLs in the HRC cohort. These instructions have been derived from instructions created by Shane McCarthy, which are available at `git://github.com/mcshane/hrc-release1.git`:

1. Download and install a revision of Htslib, Samtools, and Bcftools tagged as "rc8" from the git repository `git://github.com/mcshane/hrc-release1.git` with the following command:

   ```
   git clone --recursive git://github.com/mcshane/hrc-release1.git    \
       hrc-release1
   cd hrc-release1
   make
   ```

2. Use Samtools to pile up reads in the cohort (or multi-cohort) BAM files to generate BCF files. Only reads for samples that haplotypes were submitted for are to be used for this pileup. The pileup can be split into multiple non-overlapping chunks of the genome. The command for the pileup is

```
samtools/samtools mpileup -IE -C50 -d$MAXDEPTH -t DP,DP4 -l        \
    $SITE_LIST -g -r $CHROM:$FROM-$TO -f $REF -b $BAMLIST >        \
    $CHROM:$FROM-$TO.$COHORT.bcf
bcftools/bcftools index $CHROM:$FROM-$TO.$COHORT.bcf
```

, where `$CHROM`, `$FROM`, and `$END` are the chromosome, start, and end position, respectively, of the chunk of the genome to process. `$REF` is the GRCh37 human reference FASTA file that the reads on the BAM files are aligned to. `$MAXDEPTH` is the maximum depth per bam file, and should be set to 10 times the highest per-sample average sequencing coverage of any of the input BAM files. `$BAMLIST` is a file with the location of every input BAM file, with one location per line. `$SITE_LIST` is the MAC2 site list, which is used here to define what sites to call GLs at.

3. Concatenate all the chunks from each chromosome into a single per-chromosome VCF using `bcftools concat`:

```
bcftools/bcftools concat -Ob -f $CHROM.$COHORT.concat.list >       \
    $CHROM.$COHORT.bcf
bcftools/bcftools index $CHROM.$COHORT.bcf
```

, where `$CHROM.$COHORT.concat.list` is a file containing all the chunk BCFs generated in step 2 for a specific chromosome and cohort, with one file per line.

4. Compare the sample names of the GL BCFs with the sample names of the haplotype VCFs submitted by each project using a command like this:

```
bcftools/bcftools query -l $CHROM.$COHORT.bcf
```

If the names differ, then either fix the names or create a map from the GL BCFs to the names of the haplotype VCFs. Fixing the names in the header can be performed using

```
bcftools/bcftools view -h $CHROM.$COHORT.bcf > old_header.vcf
# map sample names to make new header and
```

```
# save in new_header.vcf
(cat new_header.vcf ; bcftools/bcftools view -H $CHROM.$COHORT.bcf) \
    | bcftools/bcftools view -Ob > $CHROM.$COHORT.new_header.bcf
```

The `samtools mpileup` command in step 2 deserves a bit more explanation. The options in order of appearance on the command line have the following meaning:

`-I` Skip indels

`-E` Recalculate the BAQ score on the fly. The BAQ is a phred-scaled probability of a read base being misaligned.

`-C50` Sets the coefficient c for downgrading the mapping quality q of reads containing an excessive number of mismatches to a new quality score $q' = c \cdot \sqrt{(c-q)/c}$, where $c = 50$. The Samtools documentation recommends this value for reads aligned with the aligner BWA.

`-d $MAXDEPTH` Read no more than `$MAXDEPTH` reads per BAM file.

`-t DP,DP4` Output the number of high-quality bases (`DP`), and the number of high-quality ref-forward, ref-reverse, alt-forward and alt-reverse bases (`DP4`).

`-l $LIST` `$LIST` is the site list that determines at which sites to perform the pileup.

`-g` Output to BCF2 format[1].

`-r $CHROM:$FROM-$TO` Only perform a pileup within the chunk defined by `$CHROM:$FROM-$TO`.

`-f $REF` Use the reference FASTA file stored at `$REF`.

`-b $BAMLIST` Pile up the reads stored in the BAM files listed in the file `$BAMLIST`.

The completed cohort likelihood BCFs were submitted to Shane McCarthy who performed the final merging using the commands

[1] All references to the BCF format in this thesis are references to the BCF2 format.

```
bcftools/bcftools norm -Ou -m+ wgs.union.sites.breakMultiAC2.vcf.gz |    \
    bcftools/bcftools query -f '%CHROM\t%POS\t%REF,%ALT\n' |             \
    htslib/bgzip -c > wgs.union.sites.AC2.alleles.gz
htslib/tabix -p vcf wgs.union.sites.AC2.alleles.gz
bcftools/bcftools merge -Ou -i DP:sum,I16:sum,QS:sum $CHROM.*.bcf |     \
    bcftools/bcftools call -Ou -mAC alleles -f GQ,GP -T                 \
    wgs.union.sites.AC2.alleles.gz | bcftools/bcftools norm -Ob -m- >   \
    $CHROM.bcf
```

These instructions perform the following actions:

1. join multi-allelic SNPs that are represented as multiple VCF lines into a single VCF line and store the resulting VCF as a tab delimited list of sites.

2. index that site list

3. For each chromosome, merge the GL BCFs into a single BCF while summing the `DP`, `I16`, and `QS` VCF `INFO` fields. Then perform multi-allelic-aware genotype calling while also outputting genotype probabilities.

4. Split multi-allelic VCF lines back into multiple bi-allelic VCF lines.

Appendix B

Extra GL clustering plots

Region	Chromosome	Start	End
1	20	16,898,517	16,955,228
2	20	24,987,790	25,044,409
3	20	59,746,593	59,79,5868
4	20	61,838,636	61,887,068
5	20	62,549,994	62,606,143

Table B.1: *Genomic coordinates of regions used for testing GL phasing program run times in Figure 3.3.*

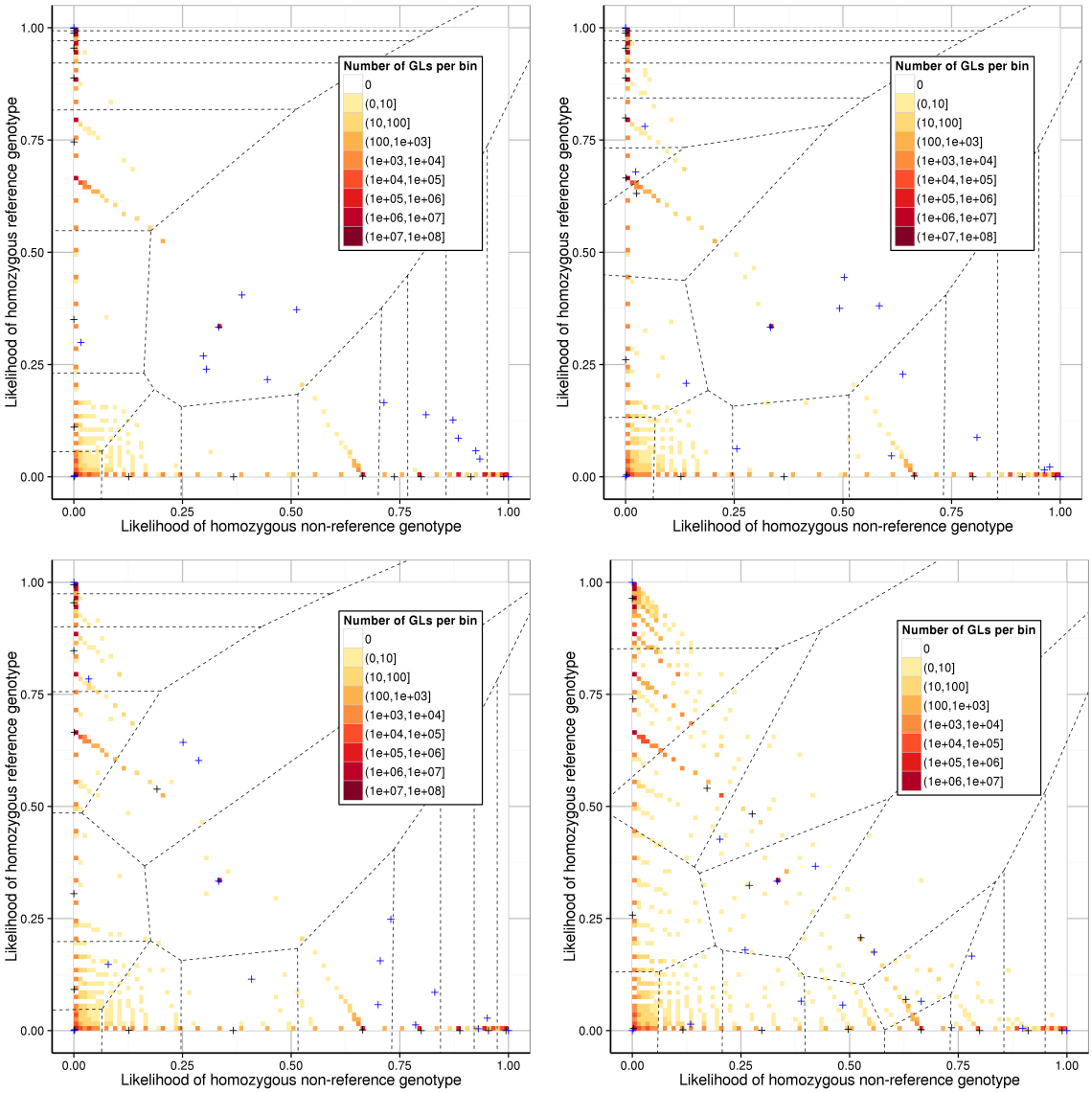

Figure B.1: *Normalized GLs of regions 1-3, and 5 in Table B.1 with results of kmeans clustering.* Regions 1-3, and 5 are top left, top right, bottom left and bottom right, respectively. GLs have been binned into a grid of 100x100 squares for presentation purposes. The number of GLs in each bin is represented as a color, which changes on a logarithmic scale. Blue crosses are initial centroids, and black crosses represent locations of centroids after clustering. Dashed lines indicate the Voronoi cells defined by post-clustering centroids. In step 3 of the GL clustering process (see section 3.6) all GLs inside a Voronoi cell are coded as the centroid of that cell.

Appendix C

VCF pre-processing for upload to the phasing server

This script can be used to pre-process a VCF so that it can be uploaded to the phasing server described in chapter 5. This script can also be downloaded from `https://phasingserver.stats.ox.ac.uk/static_pages/check_vcf.sh`.

```
#!/bin/bash
# exit on non-zero exit status
set -e

bcftools view geno.vcf.gz >/dev/null

bcftools view geno.vcf.gz -Oz -o out.vcf.gz
mv -f out.vcf.gz geno.vcf.gz

bcftools index geno.vcf.gz

bcftools view -m2 -M2 -v snps,indels geno.vcf.gz -Oz -o out.vcf.gz
mv -f out.vcf.gz geno.vcf.gz

bcftools annotate -x FORMAT geno.vcf.gz -Oz -o out.vcf.gz
mv -f out.vcf.gz geno.vcf.gz

bcftools view -g ^miss geno.vcf.gz -Oz -o out.vcf.gz
mv -f out.vcf.gz geno.vcf.gz
```

```
bcftools index -f geno.vcf.gz
bcftools index -f var_list.vcf.gz
bcftools isec -n=2 -w1 geno.vcf.gz var_list.vcf.gz -Oz -o out.vcf.gz
mv -f out.vcf.gz geno.vcf.gz

bcftools index -f geno.vcf.gz
for i in $(seq 1 22); do
  bcftools view -G -r $i geno.vcf.gz >/dev/null
done

echo "ALL CHECKS OK"
```

Bibliography

APA (1994). *Diagnostic and statistical manual of mental disorders (4th ed.)*

Boomsma, D. I., G. Willemsen, P. F. Sullivan, P. Heutink, P. Meijer, D. Sondervan, C. Kluft, G. Smit, W. a. Nolen, F. G. Zitman, J. H. Smit, W. J. Hoogendijk, R. van Dyck, E. J. C. de Geus, and B. W. J. H. Penninx (2008). Genome-wide association of major depression: description of samples for the GAIN Major Depressive Disorder Study: NTR and NESDA biobank projects. *European Journal of Human Genetics : EJHG* 16.3, pp. 335–42. DOI: 10.1038/sj.ejhg.5201979.

Browning, B. L. and S. R. Browning (2007). Efficient multilocus association testing for whole genome association studies using localized haplotype clustering. *Genetic Epidemiology* 31.5, pp. 365–375. DOI: 10.1002/gepi.20216.

Browning, B. L. and S. R. Browning (2009). A Unified Approach to Genotype Imputation and Haplotype-Phase Inference for Large Data Sets of Trios and Unrelated Individuals. *The American Journal of Human Genetics* 84.2, pp. 210–223. DOI: 10.1016/j.ajhg.2009.01.005.

Browning, B. L. and Z. Yu (2009). Simultaneous genotype calling and haplotype phasing improves genotype accuracy and reduces false-positive associations for genome-wide association studies. *The American Journal of Human Genetics* 85.6, pp. 847–861. DOI: 10.1016/j.ajhg.2009.11.004.

Browning, S. R. and B. L. Browning (2007). Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *American Journal of Human Genetics* 81.5, pp. 1084–97. DOI: 10.1086/521987.

Browning, S. R. and B. L. Browning (2011). Haplotype phasing: existing methods and new developments. *Nature Reviews Genetics* 12.10, pp. 703–714. DOI: 10.1038/nrg3054.

Cawley, S. L. and L. Pachter (2003). HMM sampling and applications to gene finding and alternative splicing. *Bioinformatics* 19.Suppl 2, pp. ii36–ii41. DOI: 10.1093/bioinformatics/btg1057.

Clark, a. G. (1990). Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution* 7.2, pp. 111–22. DOI: 10.1089/10665270152530863.

Collins, F. S., M. S. Guyer, and A. Chakravarti (1997). Variations on a theme: cataloging human DNA sequence variation. *Science* 278, pp. 1580–1. DOI: 10.1126/science.278.5343.1580.

CONVERGE Consortium (2015). Sparse whole-genome sequencing identifies two loci for major depressive disorder. *Nature* 523.7562, p. 588. DOI: 10.1038/nature14659.

Danecek, P., S. Schiffels, and R. Durbin (2014). *Multiallelic calling model in bcftools (-m)*.

Delaneau, O., B. Howie, A. J. Cox, J.-F. Zagury, and J. Marchini (2013). Haplotype estimation using sequencing reads. *American Journal of Human Genetics* 93.4, pp. 687–96. DOI: 10.1016/j.ajhg.2013.09.002.

Delaneau, O. and J. Marchini (2014). Integrating sequence and array data to create an improved 1000 Genomes Project haplotype reference panel. *Nature Communications* 5, p. 3934. DOI: 10.1038/ncomms4934.

Delaneau, O., J. Marchini, and J.-F. Zagury (2011). A linear complexity phasing method for thousands of genomes. *Nature Methods* 9.2, pp. 179–181. DOI: 10.1038/nmeth.1785.

Delaneau, O., J.-F. Zagury, and J. Marchini (2013). Improved whole-chromosome phasing for disease and population genetic studies. *Nature Methods* 10.1, pp. 5–6. DOI: 10.1038/nmeth.2307.

Department of Health Statistics and Information Systems (2013). WHO methods and data sources for global burden of disease estimates 2000-2011. *World Health Organisation, Geneva* November.

DePristo, M. a., E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. a. Philippakis, G. del Angel, M. a. Rivas, M. Hanna, A. McKenna, T. J. Fennell, A. M. Kernytsky, A. Y. Sivachenko, K. Cibulskis, S. B. Gabriel, D. Altshuler, and M. J. Daly (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics* 43.5, pp. 491–8. DOI: 10.1038/ng.806.

Dewey, F. E., M. E. Grove, C. Pan, B. A. Goldstein, J. A. Bernstein, H. Chaib, J. D. Merker, R. L. Goldfeder, G. M. Enns, S. P. David, N. Pakdaman, K. E. Ormond, C. Caleshu, K. Kingham, T. E. Klein, M. Whirl-Carrillo, K. Sakamoto, M. T. Wheeler, A. J. Butte, J. M. Ford, L. Boxer, J. P. A. Ioannidis, A. C. Yeung, R. B. Altman, T. L. Assimes, M. Snyder, E. A. Ashley, and T. Quertermous (2014). Clinical Interpretation and Implications of Whole-Genome Sequencing. *JAMA* 311.10, p. 1035. DOI: 10.1001/jama.2014.1717.

Dierks, T. and E. Rescorla (2008). RFC 5246 - The transport layer security (TLS) protocol - Version 1.2. *Network Working Group, IETF*, pp. 1–105.

Durbin, R., S. Eddy, A. Krogh, and G. Mitchison (1998). *Biological sequence analysis*, p. 356. DOI: 10.1017/CBO9780511790492.

Durbin, R. (2014). Efficient haplotype matching and storage using the positional Burrows-Wheeler transform (PBWT). *Bioinformatics (Oxford, England)* 30.9, pp. 1266–72. DOI: 10.1093/bioinformatics/btu014.

Durumeric, Z., J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson (2014). The Matter of Heartbleed. *Proceedings of the 2014 Conference on Internet Measurement Conference*, pp. 475–488. DOI: 10.1145/2663716.2663755.

Excoffier, L. and M. Slatkin (1995). Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution* 12.5, pp. 921–7.

Fearnhead, P. and P. Donnelly (2001). Estimating recombination rates from population genetic data. *Genetics* 159, pp. 1299–1318. DOI: 10.1038/nrg1227.

Fielding, R., J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee (1999). *RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1*. DOI: http://www.ietf.org/rfc/rfc2616.txt.

Fuchsberger, C., G. R. Abecasis, and D. a. Hinds (2014). Minimac2: Faster Genotype Imputation. *Bioinformatics* 31.October 2014, pp. 782–784. DOI: 10.1093/bioinformatics/btu704.

Genome of the Netherlands Consortium (2014). Whole-genome sequence variation, population structure and demographic history of the Dutch population. *Nature Genetics* 46.8, pp. 818–825. DOI: 10.1038/ng.3021.

Giardine, B., C. Riemer, and R. Hardison (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome Research* 15, pp. 1451–1455. DOI: 10.1101/gr.4086505..

Gray, R. M. (1984). Vector Quantization. *ASSP Magazine, IEEE* 1.2, pp. 4–29. DOI: 10.1109/MASSP.1984.1162229.

Howie, B. N., P. Donnelly, and J. Marchini (2009). A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genetics* 5.6, e1000529. DOI: 10.1371/journal.pgen.1000529.

Howie, B., C. Fuchsberger, M. Stephens, J. Marchini, and G. R. Abecasis (2012). Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. *Nature Genetics* 44.8, pp. 955–9. DOI: 10.1038/ng.2354.

Howie, B., J. Marchini, M. Stephens, and A. Chakravarti (2011). Genotype imputation with thousands of genomes. *G3: Genes|Genomes|Genetics* 1.6, pp. 457–70. DOI: 10.1534/g3.111.001198.

Huang, J., B. Howie, S. McCarthy, Y. Memari, K. Walter, J. L. Min, P. Danecek, G. Malerba, E. Trabetti, H.-F. Zheng, UK10K Consortium, G. Gambaro, J. B. Richards, R. Durbin, N. J. Timpson, J. Marchini, and N. Soranzo (2015). Improved imputation of low-frequency and rare variants using the UK10K haplotype reference panel. *Nature Communications* 6, p. 8111. DOI: 10.1038/ncomms9111.

Hubbard, T., D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, L. Huminiecki, A. Kasprzyk, H. Lehvaslaiho, P. Lijnzaad, C. Melsopp, E. Mongin, R. Pettett, M. Pocock, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and M. Clamp (2002). The Ensembl genome database project. *Nucleic Acids Research* 30.1, pp. 38–41. DOI: 10.1093/nar/30.1.38.

Kent, W. J., C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler (2002). The human genome browser at UCSC. *Genome Research* 12.6, pp. 996–1006. DOI: 10.1101/gr.229102.Articlepublishedonlinebeforeprintin May2002.

Khoury, M. J. and Q. Yang (1998). The future of genetic studies of complex human diseases: an epidemiologic perspective. *Epidemiology* 9.3, pp. 350–4.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science* 220.4598, pp. 671–80. DOI: 10.1126/science.220.4598.671.

Kohli, M. a., S. Lucae, P. G. Saemann, M. V. Schmidt, A. Demirkan, K. Hek, D. Czamara, M. Alexander, D. Salyakina, S. Ripke, D. Hoehn, M. Specht, A. Menke, J. Hennings, A. Heck, C. Wolf, M. Ising, S. Schreiber, M. Czisch, M. B. Müller, M. Uhr, T. Bettecken, A. Becker, J. Schramm, M. Rietschel, W. Maier, B. Bradley, K. J. Ressler, M. M. Nöthen, S. Cichon, I. W. Craig, G. Breen, C. M. Lewis, A. Hofman, H. Tiemeier, C. M. van Duijn, F. Holsboer, B. Müller-Myhsok, and E. B. Binder (2011). The neuronal transporter gene SLC6A15 confers risk to major depression. *Neuron* 70.2, pp. 252–65. DOI: 10.1016/j.neuron.2011.04.005.

Kong, A., G. Masson, M. L. Frigge, A. Gylfason, P. Zusmanovich, G. Thorleifsson, P. I. Olason, A. Ingason, S. Steinberg, T. Rafnar, P. Sulem, M. Mouy, F. Jonsson, U. Thorsteinsdottir, D. F. Gudbjartsson, H. Stefansson, and K. Stefansson (2008). Detection of sharing by descent, long-range phasing and haplotype imputation. *Nature Genetics* 40.9, pp. 1068–75. DOI: 10.1038/ng.216.

Lander, E. S., L. M. Linton, et al. (2001). Initial sequencing and analysis of the human genome. *Nature* 409.6822, pp. 860–921. DOI: 10.1038/35057062. arXiv: 11237011.

Lander, E. S. and N. J. Schork (1994). Genetic dissection of complex traits. *Science* 265.5181, pp. 2037–48. DOI: 10.1126/science.8091226.

Lewis, C. M., M. Y. Ng, A. W. Butler, S. Cohen-Woods, R. Uher, K. Pirlo, M. E. Weale, A. Schosser, U. M. Paredes, M. Rivera, N. Craddock, M. J. Owen, L. Jones, I. Jones, A. Korszun, K. J. Aitchison, J. Shi, J. P. Quinn, A. Mackenzie, P. Vollenweider, G. Waeber, S. Heath, M. Lathrop, P. Muglia, M. R. Barnes, J. C. Whittaker, F. Tozzi, F. Holsboer, M. Preisig, A. E. Farmer, G. Breen, I. W. Craig, and P. McGuffin (2010). Genome-wide association study of major recurrent depression in the U.K. population. *The American Journal of Psychiatry* 167.8, pp. 949–57. DOI: 10.1176/appi.ajp.2010.09091380.

Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* 27.21, pp. 2987–2993. DOI: 10.1093/bioinformatics/btr509. arXiv: 1203.6372.

Li, H., B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25.16, pp. 2078–2079. DOI: 10.1093/bioinformatics/btp352.

Li, N. and M. Stephens (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics* 165.4, pp. 2213–33.

Li, Y., C. Sidore, H. M. Kang, M. Boehnke, and G. R. Abecasis (2011). Low-coverage sequencing: implications for design of complex trait association studies. *Genome Research* 21.6, pp. 940–51. DOI: 10.1101/gr.117259.110.

Li, Y., C. J. Willer, J. Ding, P. Scheet, and G. R. Abecasis (2010). MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic Epidemiology* 34.8, pp. 816–34. DOI: 10.1002/gepi.20533.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28.2, pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

Loh, P.-R., P. F. Palamara, and A. L. Price (2015). *Fast and accurate long-range phasing and imputation in a UK Biobank cohort*. Tech. rep., p. 028282. DOI: 10.1101/028282.

Lunter, G. and M. Goodson (2011). Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Research* 21.6, pp. 936–9. DOI: 10.1101/gr.111120.110.

Marchini, J., D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. S. Qin, H. M. Munro, G. R. Abecasis, and P. Donnelly (2006). A Comparison of Phasing Algorithms for Trios and Unrelated Individuals. *The American Journal of Human Genetics* 78.3, pp. 437–450. DOI: 10.1086/500808.

Marchini, J. and B. Howie (2010). Genotype imputation for genome-wide association studies. *Nature Reviews Genetics* 11.7, pp. 499–511. DOI: 10.1038/nrg2796.

Marchini, J., B. Howie, S. Myers, G. McVean, and P. Donnelly (2007). A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics* 39.7, pp. 906–13. DOI: 10.1038/ng2088.

McCarthy, S., S. Das, W. Kretzschmar, R. Durbin, G. Abecasis, and J. Marchini (2015). *A reference panel of 64,976 haplotypes for genotype imputation*. Tech. rep. DOI: 10.1101/035170.

McKenna, A., M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. a. DePristo (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20.9, pp. 1297–303. DOI: 10.1101/gr.107524.110.

McVean, G. A. T. and N. J. Cardin (2005). Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 360.1459, pp. 1387–93. DOI: 10.1098/rstb.2005.1673.

Menelaou, A. and J. Marchini (2013). Genotype calling and phasing using next-generation sequencing reads and a haplotype scaffold. *Bioinformatics (Oxford, England)* 29.1, pp. 84–91. DOI: 10.1093/bioinformatics/bts632.

Morris, A. P. et al. (2012). Large-scale association analysis provides insights into the genetic architecture and pathophysiology of type 2 diabetes. *Nature Genetics* 44.9, pp. 981–90. DOI: 10.1038/ng.2383.

Muglia, P., F. Tozzi, N. W. Galwey, C. Francks, R. Upmanyu, X. Q. Kong, a. Antoniades, E. Domenici, J. Perry, S. Rothen, C. L. Vandeleur, V. Mooser, G. Waeber, P. Vollenweider, M. Preisig, S. Lucae, B. Müller-Myhsok, F. Holsboer, L. T. Middleton, and a. D. Roses (2010). Genome-wide association study of recurrent major depressive disorder in two European case-control cohorts. *Molecular Psychiatry* 15.6, pp. 589–601. DOI: 10.1038/mp.2008.131.

Nikpay, M. et al. (2015). A comprehensive 1,000 Genomes-based genome-wide association meta-analysis of coronary artery disease. *Nature Genetics* 47.10, pp. 1121–30. DOI: 10.1038/ng.3396.

O'Connell, J., D. Gurdasani, O. Delaneau, N. Pirastu, S. Ulivi, M. Cocca, M. Traglia, J. Huang, J. E. Huffman, I. Rudan, R. McQuillan, R. M. Fraser, H. Campbell,

O. Polasek, G. Asiki, K. Ekoru, C. Hayward, A. F. Wright, V. Vitart, P. Navarro, J.-F. Zagury, J. F. Wilson, D. Toniolo, P. Gasparini, N. Soranzo, M. S. Sandhu, and J. Marchini (2014). A general approach for haplotype phasing across the full spectrum of relatedness. *PLoS Genetics* 10.4, e1004234. DOI: 10.1371/journal.pgen.1004234.

O'Connell, J., K. Sharp, N. Shrine, L. Wain, I. Hall, M. Tobin, J.-F. Zagury, O. Delaneau, and J. Marchini (2016). Haplotype estimation for biobank-scale data sets. *Nature Genetics* 48.7, pp. 817–20. DOI: 10.1038/ng.3583.

Parsons, D. W., A. Roy, S. E. Plon, S. Roychowdhury, and A. M. Chinnaiyan (2014). Clinical Tumor Sequencing: An Incidental Casualty of the American College of Medical Genetics and Genomics Recommendations for Reporting of Incidental Findings. *Journal of Clinical Oncology* 32.21, pp. 2203–2205. DOI: 10.1200/JCO.2013.54.8917.

Pasaniuc, B., N. Rohland, P. J. McLaren, K. Garimella, N. Zaitlen, H. Li, N. Gupta, B. M. Neale, M. J. Daly, P. Sklar, P. F. Sullivan, S. Bergen, J. L. Moran, C. M. Hultman, P. Lichtenstein, P. Magnusson, S. M. Purcell, D. W. Haas, L. Liang, S. Sunyaev, N. Patterson, P. I. W. de Bakker, D. Reich, and A. L. Price (2012). Extremely low-coverage sequencing and imputation increases power for genome-wide association studies. *Nature Genetics* 44.6, pp. 631–635. DOI: 10.1038/ng.2283.

Pritchard, J. K. and M. Przeworski (2001). Linkage disequilibrium in humans: models and data. *American Journal of Human Genetics* 69.1, pp. 1–14. DOI: 10.1086/321275.

Rabiner, L. (1989). a tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*.

Rietschel, M., M. Mattheisen, J. Frank, J. Treutlein, F. Degenhardt, R. Breuer, M. Steffens, D. Mier, C. Esslinger, H. Walter, P. Kirsch, S. Erk, K. Schnell, S. Herms, H.-E. Wichmann, S. Schreiber, K.-H. Jöckel, J. Strohmaier, D. Roeske, B. Haenisch, M. Gross, S. Hoefels, S. Lucae, E. B. Binder, T. F. Wienker, T. G. Schulze, C. Schmäl, A. Zimmer, D. Juraeva, B. Brors, T. Bettecken, A. Meyer-Lindenberg, B. Müller-Myhsok, W. Maier, M. M. Nöthen, and S. Cichon (2010). Genome-wide association-, replication-, and neuroimaging study implicates HOMER1 in the etiology of major depression. *Biological Psychiatry* 68.6, pp. 578–85. DOI: 10.1016/j.biopsych.2010.05.038.

Ripke, S. et al. (2013). A mega-analysis of genome-wide association studies for major depressive disorder. *Molecular Psychiatry* 18.4, pp. 497–511. DOI: 10.1038/mp.2012.21.

Scheet, P. and M. Stephens (2006). A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *American Journal of Human Genetics* 78.4, pp. 629–644. DOI: 10.1086/502802.

Sharp, K., W. Kretzschmar, O. Delaneau, and J. Marchini (2016). Phasing for medical sequencing using rare variants and large haplotype reference panels. *Bioinformatics* 32.13, pp. 1974–80. DOI: 10.1093/bioinformatics/btw065.

Shyn, S. I., J. Shi, J. B. Kraft, J. B. Potash, J. a. Knowles, M. M. Weissman, H. a. Garriock, J. S. Yokoyama, P. J. McGrath, E. J. Peters, W. a. Scheftner, W. Coryell, W. B. Lawson, D. Jancic, P. V. Gejman, a. R. Sanders, P. Holmans, S. L. Slager, D. F. Levinson, and S. P. Hamilton (2011). Novel loci for major depression identified by genome-wide association study of Sequenced Treatment Alternatives to Relieve Depression and meta-analysis of three studies. *Molecular Psychiatry* 16.2, pp. 202–15. DOI: 10.1038/mp.2009.125.

Stephens, M., N. J. Smith, and P. Donnelly (2001). A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics* 68.4, pp. 978–989. DOI: 10.1086/319501.

Stephens, M. and P. Donnelly (2000). Inference in molecular population genetics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62.4, pp. 605–635. DOI: 10.1111/1467-9868.00254.

Stephens, M. and P. Donnelly (2003). A comparison of bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics* 73.5, pp. 1162–9. DOI: 10.1086/379378.

Stephens, M. and P. Scheet (2005). Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *American Journal of Human Genetics* 76, pp. 449–462. DOI: 10.1086/428594.

Subashini, S. and V. Kavitha (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications* 34.1, pp. 1–11. DOI: 10.1016/j.jnca.2010.07.006.

Sullivan, P. F., E. J. C. de Geus, G. Willemsen, M. R. James, J. H. Smit, T. Zandbelt, V. Arolt, B. T. Baune, D. Blackwood, S. Cichon, W. L. Coventry, K. Domschke, a. Farmer, M. Fava, S. D. Gordon, Q. He, a. C. Heath, P. Heutink, F. Holsboer, W. J. Hoogendijk, J. J. Hottenga, Y. Hu, M. Kohli, D. Lin, S. Lucae, D. J. Macintyre, W. Maier, K. a. McGhee, P. McGuffin, G. W. Montgomery, W. J. Muir, W. a. Nolen, M. M. Nöthen, R. H. Perlis, K. Pirlo, D. Posthuma, M. Rietschel, P. Rizzu, a. Schosser, a. B. Smit, J. W. Smoller, J.-Y. Tzeng, R. van Dyck, M. Verhage, F. G. Zitman, N. G. Martin, N. R. Wray, D. I. Boomsma, and B. W. J. H. Penninx (2009). Genome-wide association for major depressive disorder: a possible role for the presynaptic protein piccolo. *Molecular Psychiatry* 14.4, pp. 359–75. DOI: 10.1038/mp.2008.125.

Sullivan, P. F., M. C. Neale, and K. S. Kendler (2000). Genetic epidemiology of major depression: review and meta-analysis. *The American Journal of Psychiatry* 157.10, pp. 1552–62. DOI: 10.1176/appi.ajp.157.10.1552.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature* 467.7319, pp. 1061–73. DOI: 10.1038/nature09534.

The 1000 Genomes Project Consortium (2012). An integrated map of genetic variation from 1,092 human genomes. *Nature* 491.7422, pp. 56–65. DOI: 10.1038/nature11632.

The 1000 Genomes Project Consortium (2015). A global reference for human genetic variation. *Nature* 526.7571, pp. 68–74. DOI: 10.1038/nature15393.

The International HapMap Consortium (2003). The International HapMap Project. *Nature* 426.6968, pp. 789–796. DOI: 10.1038/nature02168.

The International HapMap Consortium (2005). A haplotype map of the human genome. *Nature* 437.7063, pp. 1299–1320. DOI: 10.1038/nature04226.

The International HapMap Consortium (2007). A second generation human haplotype map of over 3.1 million SNPs. *Nature* 449.7164, pp. 851–61. DOI: 10.1038/nature06258.

The International HapMap Consortium (2010). Integrating common and rare genetic variation in diverse human populations. *Nature* 467.7311, pp. 52–8. DOI: 10.1038/nature09298.

The International SNP Map Working Group (2001). A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature* 409.6822, pp. 928–933. DOI: 10.1038/35057149.

The Wellcome Trust Case Control Consortium (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 447.7145, pp. 661–78. DOI: 10.1038/nature05911. arXiv: t8jd4qr3m [13960].

UK10K Consortium (2015). The UK10K project identifies rare variants in health and disease. *Nature* 526.7571, pp. 82–90. DOI: 10.1038/nature14962.

Vaidya, P. M. (1989). An O(n log n) algorithm for the all-nearest-neighbors Problem. *Discrete & Computational Geometry* 4.1, pp. 101–115. DOI: 10.1007/BF02187718.

Venter, J. C. et al. (2001). The sequence of the human genome. *Science (New York, N.Y.)* 291.5507, pp. 1304–51. DOI: 10.1126/science.1058040.

Wang, Y., J. Lu, J. Yu, R. a. Gibbs, and F. Yu (2013). An integrative variant analysis pipeline for accurate genotype/haplotype inference in population NGS data. *Genome Research* 23.5, pp. 833–42. DOI: 10.1101/gr.146084.112.

Watterson, G. a. (1975). On the number of segregating sites in genetical models without recombination. *Theoretical Population Biology* 7, pp. 256–276. DOI: 10.1016/0040-5809(75)90020-9.

Welter, D., J. MacArthur, J. Morales, T. Burdett, P. Hall, H. Junkins, A. Klemm, P. Flicek, T. Manolio, L. Hindorff, and H. Parkinson (2014). The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research* 42.Database issue, pp. D1001–6. DOI: 10.1093/nar/gkt1229.

Wetterstrand, K. A. (2013). *DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program.*

Williams, A. L., N. Patterson, J. Glessner, H. Hakonarson, and D. Reich (2012). Phasing of many thousands of genotyped samples. *American Journal of Human Genetics* 91.2, pp. 238–51. DOI: 10.1016/j.ajhg.2012.06.013.

Wray, N. R., M. L. Pergadia, D. H. R. Blackwood, B. W. J. H. Penninx, S. D. Gordon, D. R. Nyholt, S. Ripke, D. J. MacIntyre, K. a. McGhee, a. W. Maclean, J. H. Smit, J. J. Hottenga, G. Willemsen, C. M. Middeldorp, E. J. C. de Geus, C. M. Lewis, P. McGuffin, I. B. Hickie, E. J. C. G. van den Oord, J. Z. Liu, S. Macgregor, B. P. McEvoy, E. M. Byrne, S. E. Medland, D. J. Statham, a. K. Henders, a. C. Heath, G. W. Montgomery, N. G. Martin, D. I. Boomsma, P. a. F. Madden, and P. F. Sullivan (2012). Genome-wide association study of major depressive disorder: new

results, meta-analysis, and lessons learned. *Molecular Psychiatry* 17.1, pp. 36–48. DOI: 10.1038/mp.2010.109.