Statistical Prediction of HLA Alleles and Relatedness Analysis in Genome-Wide Association Studies

Xiuwen Zheng

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2013

Reading Committee:

Bruce S. Weir, Chair

Ellen M. Wijsman

Sharon R. Browning

Program Authorized to Offer Degree: Biostatistics

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my dissertation advisor, Bruce S. Weir, for his inspiration and guidance. I would also like to thank the members of my supervisory committee, Ellen M. Wijsman, Sharon R. Browning and Jonathan C. Wakefield for their inputs into this dissertation.

DEDICATION

I lovingly dedicate this dissertation work to my family, who supported me each step of the way.

University of Washington

Abstract

Statistical Prediction of HLA Alleles and Relatedness Analysis in Genome-Wide Association Studies

Xiuwen Zheng

Chair of the Supervisory Committee:
Professor Dr. Bruce S. Weir
Chair's department

Genome-wide association studies (GWAS) have been used widely in the last decade to investigate the genetic basis of many complex diseases and traits. However, the significantly associated SNPs have explained relatively little of the heritability of most common diseases and traits, and most of these SNPs give small increments in risk or have small effect sizes. This phenomenon has led to a well-known problem of "missing heritability" of complex diseases and traits using GWAS. Discussions of "missing heritability" in GWAS require examination of the underlying assumption of linkage disequilibrium (LD) at the population level. The human leukocyte antigen (HLA) region has been considered as a high-LD region, but it is also known to be highly polymorphic. The study of HLA imputation could well provide us new insights into the "missing heritability", and a new method "HIBAG" is proposed that makes predictions by averaging HLA type posterior probabilities over an ensemble of classifiers built on bootstrap samples. Another major analytical factor affecting the interpretation of GWAS is cryptic relatedness and population stratification. Principal component analysis (PCA) has been widely used to detect and correct for popula-

tion structure, however it is a model-free approach and seems like a "black box". An interpretation of PCA based on identity-by-descent (IBD) measures is given, and an approximately linear transformation between the projection of individuals onto principal components and allele admixture fractions assuming two or more ancestral populations is revealed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

Genome-wide association studies (GWAS) have been used widely in the last decade to investigate the genetic basis of many complex diseases and traits. The aim of GWAS is to detect associations between genetic variants, such as common single-nucleotide polymorphisms (SNPs) and common diseases, such as diabetes. A focus of GWAS has been the common disease – common variant (CD-CV) hypothesis, i.e., a complex disease is largely attributable to a moderate number of common variants [8, 89]. One of the greatest hopes for GWAS was to characterize genetic risks at the DNA level [69, 108]. GWAS publications now list hundreds of common variants which are statistically correlated with common illnesses and traits (`http://www.genome.gov/gwastudies/`) [45].

The significantly associated SNPs have explained relatively little of the heritability of most common diseases and traits, and most of these SNPs give small increments in risk or have small effect sizes [29, 69]. For example, for human height, a highly heritable trait, previous GWASs with thousands of individuals have identified over 100 loci associated with height, and together they explain only about 10% of the total phenotypic variance [56]. However, a traditionally estimated genetic contribution to human height is $\sim$ 80% heritability for subjects of European descent from familial studies. This phenomenon has led to a well-known problem of "missing heritability" of complex diseases and traits using GWAS. A partial explanation for this discrepancy is

that many of the variants contributing to a trait do not reach genome-wide significance levels [119].

Heritability has been used by geneticists to quantify the portion of observed variation in a trait due to genetic factors. Observed trait phenotypes have often been conceptualized by partitioning in a linear model:

$$\text{Phenotype(P)} = \text{Genotype(G)} + \text{Environment(E)} \tag{1.1}$$

A more nuanced version is:

$$\text{Phenotype(P)} = \text{Genotype(G)} + \text{Environment(E)} + \text{G} \times \text{E Interactions} \tag{1.2}$$

Formally, broad-sense heritability is defined as a ratio of variance components, the variance of genotypic values over the total phenotypic variance. It has been estimated traditionally from regression of offspring on parental phenotypes, or from the correlation of sibs, or from the difference in correlations of monozygotic and dizygotic twin pairs [52, 109].

The reasons for "missing heritability" have been hotly debated during the last five years [29, 38, 67, 69, 71, 108]. The pertinent points of different arguments range from epigenetic and environmental factors: DNA sequences not playing an important role in the risk to common disease [67, 101], to the "common disease-many rare alleles" model [71], infinitesimal model (many variants of small effect) [38], and structural variations [29]. Heritabilities estimated by quantitative genetic approaches using all SNPs simultaneously were closer to traditional estimates: Yang et al. (2010) reported that 45% of variance in human height can be explained by taking 294,831 SNPs into account [118].

1.1 Heritability and HLA Imputation

Discussions of "missing heritability" in GWAS require examination of the underlying assumption of linkage disequilibrium (LD) at the population level. LD, also referred to as gametic phase disequilibrium, is defined as the non-random association of alleles at two or more loci on haplotypes. Factors such as genetic linkage, selection, the rate of recombination, the rate of mutation, genetic drift, non-random mating, and population structure, could all influence the level of observed LD [108]. The rate of decay of LD as a function of chromosomal distance between pairs of loci could be used to design a relatively small number of tagging markers across the whole genome, since one variant could be a good surrogate for other variants if they are in a high-LD block. This was the logic leading to the establishment of the HapMap Project [50, 104]. It was thought that a selection of $\sim 500,000$ common SNPs was sufficient to tag common variation in non-African populations, although over 10 million common DNA variants, primarily SNPs, exist [50].

Linkage disequilibrium was described for genetic markers in the Human Leukocyte Antigen (HLA) region on chromosome 6 by Dausset et al. in 1978 [24, 107]. The HLA region has an important role in the immune and autoimmune system, and it is been known to be highly polymorphic. Some of what we know today about genetic variation, including haplotypic structure [77], was first discovered from studies of this region. In serological studies, the term "haplotype" referred to the combination of individual antigenic allelic determinants. Sperm typing was first applied to the HLA region to identify and define recombination hotspots [107]. The HLA complex has been at the forefront of human genetic research, but still remains an evolutionary puzzle: little is known about the mechanisms maintaining polymorphism and LD [31].

Imputation-based HLA typing is of great interest to geneticists who work on associations between diseases and HLA alleles because of lower costs compared to sequencing-based approaches. The idea of this typing method is to infer HLA type using flanking SNP markers rather than by directly sequencing. Based on the current nomenclature the HLA typing results refer to phenotypes, and unique serological features are required to determine a new allele. Therefore, HLA imputation could be considered as a special predictive problem of phenotypes at the population level, where the phenotype of an individual at a specific locus is controlled by the two HLA alleles which that individual carries. The issue is whether we can use fewer tagging markers to capture much larger number of common variants in GWAS that contain causal variants. Hence, the study of HLA imputation could well provide us new insights into the "missing heritability".

1.2 Population Structure and Relatedness

Another major analytical factor affecting the interpretation of GWAS is cryptic relatedness and population stratification [19, 21, 71]. The presence of a systematic difference in allele frequencies, unrleated to disease, between cases and controls could yield spurious marker-disease associations and inflate false-positive findings. The analytical challenges can be addressed by using principal component analysis (PCA) to detect and correct for population structure [83, 86] and identity-by-descent (IBD) methods to identify the degree of relatedness between each pair of study samples [21]. These popular approaches to adjustment are appropriate and necessary but might be not sufficient [71]. PCA has been widely used to detect population stratification, however it is a model-free approach and seems like a "black box", although it is known that principal component axes often represent perpendicular gradients in geographic

space [76, 81, 83]. It would be of interest to open this "black box" to reveal deeper principles.

1.3 Data Challenges

To date the large volumes of data generated by chip- and sequencing- based GWASs from thousands of study samples and millions of variants have posed significant computational challenges. In the last ten years, chip-based genotyping technologies, such as the Illumina 1M BeadChip and the Affymetrix 6.0 chip, have allowed hundreds of thousands of common variants (SNPs) across the whole genome to be scored simultaneously. The Gene, Environment Association Studies Consortium (GENEVA) has generated genotypic data using chip-based genotyping techniques, with a large number of research participants ($n > 80,000$) from 14 independently designed studies of various phenotypes [22]. Currently, the field of population genetics is moving from chip data to sequencing data. Next-generation sequencing techniques are being adopted to investigate common and rare variants, making the analyses of large-scale genotypic data even more challenging. For example, the 1000 Genomes Project has identified approximately 38 million single nucleotide polymorphisms (SNPs), 1.4 million short insertions and deletions, and more than 14,000 larger deletions from whole-genome sequencing technologies [2]. In the near future, new technologies, like third-generation whole-genome sequencing [30], will enable data to be generated at an unprecedented scale [96]. The computational burden associated with analyses of genome-wide variants is especially evident with large sample and variant sizes, and really requires efficient numerical implementation and data management.

1.4 Structure of this Dissertation

Chapter Two of this dissertation considers the problem of HLA genotype imputation using SNPs in GWAS, in which a new prediction method "HIBAG" is proposed. The third chapter describes a genetic framework based on relatedness measures to interpret PCA analytically, and a new approach to identify population structure is also proposed based on this framework. The fourth chapter emphasizes the importance of computing toolset for big-data analyses of genome-wide variants. A solution with three R packages from the "CoreArray" project was proposed and released. The last chapter summarizes the implication of each study and provides possible directions for future research.

Chapter 2

HLA Genotype Imputation in Genome-Wide Association Studies

2.1 Abstract

Genotyping of classical HLA alleles is an essential tool in the analysis of diseases and adverse drug reactions with associations mapping to the major histocompatibility complex (MHC). However, deriving high-resolution HLA types subsequent to whole-genome SNP typing or sequencing is often cost prohibitive for large samples. An alternative approach takes advantage of the extended haplotype structure within the MHC to predict HLA alleles using dense SNP genotypes, such as those available from genome-wide SNP panels. Current methods for HLA imputation are difficult to apply or may require the user to have access to large training data sets with SNP and HLA types. We propose HIBAG, **H**LA **I**mputation using attribute **BAG**ging, that makes predictions by averaging HLA type posterior probabilities over an ensemble of classifiers built on bootstrap samples. We assessed the performance of HIBAG using HLA data from 2,668 subjects of European ancestry with four-digit HLA and Illumina 1M SNP genotypes, divided randomly into training and validation parts. Prediction accuracies of *HLA–A*, *B*, *C*, *DRB1*, *DQA1*, *DQB1* and *DPB1* range from 94.8% to 99.2% with call rate over 90% using a set of SNP markers common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms. HIBAG performed well compared to BEAGLE and HLA*IMP. This method is implemented

in the freely-available R package that includes pre-fit classifiers based on these data, providing a readily available imputation approach without the need to have access to large training datasets.

2.2 Introduction

The human leukocyte antigen (HLA) system, located in the major histocompatibility complex (MHC) on chromosome 6p21.3, is highly polymorphic. This region has been shown to be important in human disease, adverse drug reactions and organ transplantation [99]. HLA genes play a role in the immune system and autoimmunity as they are central to the presentation of antigens for recognition by T cells. Since they have to provide defense against a great diversity of environmental microbes, HLA genes must be able to present a wide range of peptides. Evolutionary pressure at these loci have given rise to a great deal of functional diversity. For example, the *HLA–B* locus has 2,110 high-resolution alleles listed in the Nov 2012 release of the IMGT-HLA Database [94] (`http://www.ebi.ac.uk/imgt/hla/`). The ever-growing list of HLA alleles is maintained by the World Health Organization (WHO) Nomenclature Committee, with an average discovery rate of 15 new alleles per month (`http://www.imgt.org`) [36].

Classical HLA genotyping methodologies have been predominately developed for tissue typing purposes, with sequence based typing (SBT) approaches currently considered the gold standard for diagnostic purposes. While there is widespread availability of vendors offering HLA genotyping services, the complexities make using a SBT approach time-consuming and cost-prohibitive for most research studies wishing to look in detail at the involvement of classical HLA genes in disease.

The current developing methods for HLA typing include imputation-based ap-

Table 2.1: The observed number of unique HLA alleles and genotypes for each locus.

	HLA –						
	A	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
High-resolution HLA alleles in the Nov 2012 release of the IMGT-HLA Database							
# of alleles	1527	2110	1200	958	31	128	136
HLARES data of European ancestry [1]							
# of individuals	1857	2572	1866	2436	1740	1924	1624
# of unique alleles	48	88	37	55	17	21	26
# of unique genotypes	208	572	201	355	84	118	135
All HLARES data [1]							
# of individuals	3035	4131	3007	3864	2798	3150	2478
# of unique alleles	85	144	49	80	19	27	49
# of unique genotypes	378	1099	321	617	116	150	205

[1]: subjects from multiple GlaxoSmithKline clinical trials.

proaches [32]. The idea of imputation is to infer the polymorphism of HLA alleles rather than to directly sequence, and this method relies on a training set. The previous studies have suggested that the existence of some HLA alleles can be predicted by a SNP-based tagging approach [25, 33]. SNP-based tagging does not offer a definitive solution to HLA genotyping by prediction since many HLA alleles are found on multiple haplotype backgrounds [59].

However, HLA imputation is a special challenge compared to SNP imputation methods for other regions of human genome. First, the HLA region is the most gene-dense and polymorphic region of the human genome sequences. As shown in Table 2.1, it has been found there are more than one thousand protein-level alleles for the class I genes (*HLA–A*, *B* and *C*) all over the world. High resolution (4-digit) HLA typing is the focus in this study, referring to protein level. In the HLARES data, *HLA–B* locus is the most polymorphic with 144 alleles in total, and large numbers of unique genotypes are also observed. Second, recombination and linkage disequilibrium

patterns across the HLA region are not uniform [107]. The HLA has been known to have some high recombination rates like hotspots, and some of the highest amounts of LD found in the genome. The TAP2 recombination hotspot [53], located in the HLA region, is one of the best known hotspots in the human genome [74]. Structural variants resulting from non-allelic homologous recombination are apparently enriched in the HLA region from the data of the 1000 genome project [1]. Third, it is difficult to genotype accurately because of the high degree of duplicated genes and structural variants within the HLA region. The raw HLA typing results may be consistent with more than one genotype called ambiguously due to incomplete genomic coverage of sequence or inability to set phase for allele determination. HLA typing itself is also usually not uniform between labs.

Based on the current nomenclature the HLA typing results refer to phenotypes, and unique serological features are required to determine a new allele. A low-resolution HLA allele could correspond to many synonymous and non-synonymous DNA sequences. For example, *HLA–B* gene contains 8 exons. Exon 1 encodes the leader peptide, and the polymorphisms within exon 2 and exon 3 are responsible for the peptide binding specificity of each class one molecule. Low-resolution typing usually refers to exon 1 for HLA class I genes, whereas the exon 2 and exon 3 sequences are employed further to distinguish high-resolution types. However, the polymorphisms within the other exons and introns could be required for higher-resolution tissue typing.

To impute HLA types from multiple SNP markers, Leslie et al. (2008) used an identity-by-descent (IBD) model based on approximate coalescent models [61, 73] to develop their LDMhc algorithm, and used a leave-one-out cross-validation scheme for SNP selection [59]. Dilthey et al. (2010) subsequently developed integrated software

HLA*IMP for imputing classical HLA alleles from SNP genotypes based on LDMhc, with a modified SNP selection function that leads to pronounced increases in call rate [28]. A training set of SNP haplotypes with known HLA alleles are required by LDMhc, as well as a fine genetic map of the region [28]. Leslie et al. (2008) emphasized that the LDMhc algorithm relied on high-quality haplotype information, which may be obtained by inferring trio data. In addition, HLA-IBD [98] and WSG-HI [117] were also proposed to impute HLA alleles utilizing haplotypes and trio information.

However, most experimental techniques for determining SNPs provide genotypes rather than haplotypes, such as chip-based SNP genotyping. Li et al. (2011) introduced a complementary method MAGprediction with respect to LDMhc, for predicting highly polymorphic alleles using unphased SNP data as the training data set [64]. BEAGLE, an alternative imputation method to the approximate coalescent models, allows predicting multiallelic loci [15]. It locally clusters the observed haplotypes at each position, based on similarity of the haplotypes at markers in the local vicinity, with a model assumption of variable-length Markov chains [16, 14]. Recently, it was used for HLA imputation in a genetic association analysis [93]. That study illustrates how imputation of functional variation can help fine-map association signals in the HLA region.

Here we propose a new method for **H**LA **I**mputation using attribute **BAG**ging, **HIBAG**, that is highly accurate, computationally tractable, and can be used with published parameter estimates, eliminating the need to access large training samples [121]. It combines the concepts of attribute bagging, an ensemble classifier method, with haplotype inference from unphased SNPs and HLA types. Attribute bagging is a technique for improving the accuracy and stability of classifier ensembles deduced using bootstrap aggregating and random subsets of variables [9, 12, 17], as shown in

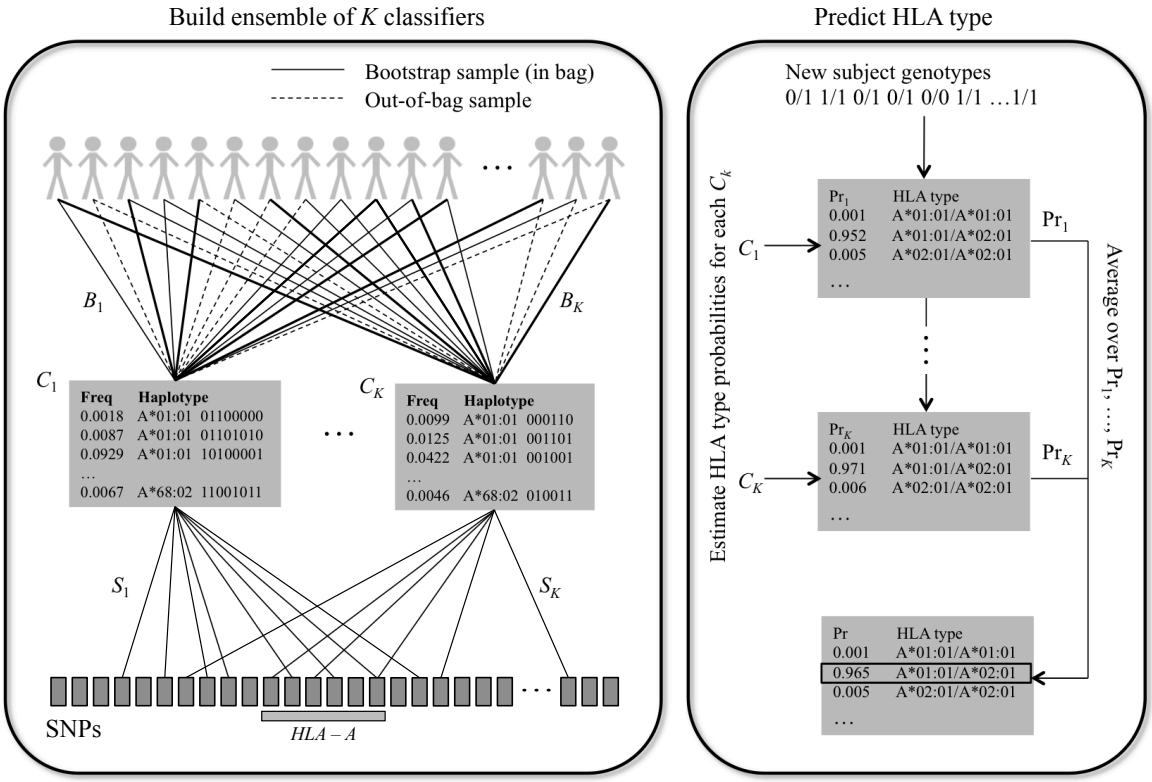


Figure 2.1: Overview of the HIBAG prediction algorithm. HIBAG is an ensemble classifier consisting of individual classifiers with HLA and SNP haplotype probabilities estimated from bootstrapped samples and SNP subsets. The SNP subsets are determined by a variable selection algorithm with a random component. HLA type predictions are averaged over the posterior probabilities from all classifiers.

Figure 2.1. Bootstrap aggregating (or bagging) helps to improve accuracies, stabilize the model and reduce data noise. In Figure 2.1, individual classifiers are created which utilize a subset of SNPs to predict HLA types and haplotype frequencies estimated from a training data set of SNPs and HLA types. Each of the classifiers employes a variable selection algorithm with a random component to determine a subset of the SNPs. HLA type predictions are determined by maximizing the average posterior probabilities from all classifiers.

The outline of this chapter is described as follows. Section 2.3 reviews the previous

sequencing- and imputation-based HLA typing methods, and Section 2.4 describes the proposed HIBAG algorithm in details. In this study, high-resolution (4-digit) HLA alleles are the focus. I investigated the overall performance of HIBAG using HLA data and SNP genotypes from HapMap [25], the British 1958 birth cohort data of the Wellcome Trust Case Control Consortium (WTCCC) [59] and HLARES from multiple GlaxoSmithKline (GSK) clinical trials in Section 2.5 and 2.6. Finally, a discussion was made in Section 2.7.

2.3 Methods of HLA Typing

An HLA gene is a DNA fragment consisting of several thousand of basepairs, for example, *HLA–B* gene contains 3,286 basepairs spanned by the 8 exons. Each allele name has a unique number corresponding to up to four sets of digits separated by colons, as shown in Figure 2.2. The length of the allele designation is dependent on the sequence of the allele and that of its nearest relative. The first two digits represent the allele family, which often corresponds to the serological antigen carried by an allotype. The third and fourth digits describe the subtypes and represents different amino acid sequences of the encoded protein. The remaining digits denote any synonymous mutations in exons and introns respectively. The suffix indicates expression level or other non-genomic data. The convention is to use a four-digit code to distinguish alleles which differ in their protein products, and we primarily investigate four-digit HLA alleles in this study. Several sequence- and imputation-based HLA typing methods are summarized in Table 2.2: sequence-specific oligonucleotide probes (SSOP), sequence specific primers (SSP), Sanger sequence based typing (SBT), next-generation sequencing (NGS), HLA*IMP [28], HLA-IBD [98], WSG-HI [117], MAGprediction [64] and BEAGLE [15]. More details are given in Section 2.3.1, 2.3.2 and 2.3.3.

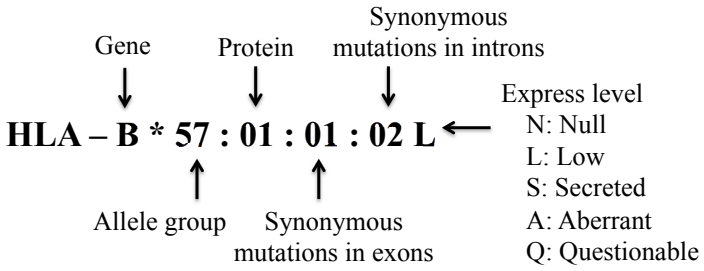


Figure 2.2: Nomenclature of HLA Alleles.

The mathematical symbols used in the remaining part of Section 2.3 and 2.4 are described ahead in Table 2.3. Let y denote an HLA allele, x be a SNP allele ($x = 1$ if it is A allele, otherwise $x = 0$) and $g \in \{0, 1, 2\}$ be a SNP genotype. An HLA type at a specified locus is denoted as an unordered pair of alleles $\langle y^{(1)}, y^{(2)} \rangle$. Let n be the total number of samples, m be the total number of candidate SNPs, and the index j $(1 \leq j \leq m)$ indicates the j^{th} SNP marker. $\{x_1, ..., x_m\}$ denotes a SNP haplotype with loci $x_1, ..., x_m$.

Table 2.2: Summary of existing DNA HLA typing methods.

Method	Characteristics	Limitedness or potential weakness
Conventional HLA typing methods		
SSOP, SSP, SBT [1]	detect genetic variants at molecular level	time-consuming, not cost-effective and ambiguous typing results
Current HLA typing method		
NGS [2]	massively parallel clonal sequencing, reduce the ambiguity problem, the time and typing cost	less cost-effective than imputations for population-scale studies
Imputation-based methods relying on haplotypes		
HLA*IMP [3]	based on approximate coalescent models, utilize a small set of selected SNPs to impute	the quality of inferred haplotypes from unrelated individuals
HLA-IBD	imputes HLA alleles by inferred IBD segments	haplotype quality
WSG-HI	utilizes a weighted graph based on a haplotype similarity measure	haplotype quality
Imputation-based methods using unphased data		
MAGprediction	infers haplotype frequencies by the estimating equation technique for likelihood function, uses a small set of selected SNPs	Ayele et al. (2012) reported that it failed to impute high-resolution alleles at *HLA–DRB1* and *DQB1* of African ancestry
BEAGLE [3]	locally cluster the observed haplotypes assuming variable-length Markov chains, and can use all SNPs across the HLA region	requires a large reference sample to produce accurate results

[1]: sequence-specific oligonucleotide probes (SSOP), sequence specific primers (SSP), and Sanger sequence based typing (SBT).
[2]: next-generation sequencing (NGS).
[3]: two leading HLA imputation methods. A comparison among HLA*IMP, BEAGLE and the proposed method (HIBAG) was made in this study.

Table 2.3: Summary of mathematical symbols used in Chapter 2.

Symbol	Description
n	the total number of samples
m	the total number of SNPs
y	an HLA allele at a specific locus of interest
$\langle y^{(1)}, y^{(2)} \rangle$	an unordered pair of HLA alleles
x	a SNP allele ($x = 1$ if it is A allele, otherwise $x = 0$)
$\{x_1, ..., x_m\}$	a SNP haplotype of loci $x_1, ..., x_m$
g	a SNP genotype ($g \in \{0, 1, 2\}$)
$\{g_1, ..., g_m\}$	a SNP genotype profile of loci $x_1, ..., x_m$
the index j	the j^{th} SNP marker, $1 \leq j \leq m$
K	the total number of individual classifiers
C_k	the k^{th} individual classifier
$\mathcal{S}_k$	the SNP subset used by the k^{th} individual classifier

2.3.1 DNA Typing Methods

Since the late 1950s, the field of Immunogenetics has been searching for a simple method that provides accurate and highly informative HLA typing. However, the extensive allelic diversity has made and continues to make high-resolution HLA DNA typing challenging [32]. Current typing methods include sequence-specific oligonucleotide probes (SSOP), sequence specific primers (SSP) and Sanger sequence based typing (SBT). SSOP and SSP are two conventional polymerase chain reaction (PCR) methods, and they were designed to detect variable sequence motifs in PCR-amplified genes. SBT enables all nucleotides to be identified within a PCR product providing the highest resolution possible for HLA genes, and is the only way to define new alleles by now [36]. However, these methods are time-consuming and not cost-effective, compared to newer technologies and methods. One great disadvantage of SBT is the fact that it could yield ambiguous typing results because of incomplete genomic coverage and inability to set phase for allele determination.

In the past few years, the next-generation sequencing (NGS) technology has been used for HLA typing, characterized by massively parallel clonal sequencing [36, 32], which offers the promise of significantly reducing the ambiguity problem. In the study of Holcomb et al. (2011), a multi-site and double-blind evaluation was conducted on the 454 Life Sciences Genome Sequencer (GS FLX) NGS, in which eight laboratory sites used the same reagents, protocols and software to sequence and assign 10 HLA genotypes for the same set of 20 DNA samples [46]. The HLA types of these 20 samples were determined by a combination of multiple conventional methods, including SBT and SSP. Of the 1280 genotypes considered, the call rate is 95%, and concordance with called genotypes ranged from 95.3% to 99.4% with overall concordance of 97.2% between NGS and conventional methods.

One new approach to HLA typing is to infer the polymorphism rather than to directly sequence [32]. Leslie et al. (2008) firstly proposed their LDMhc imputation algorithm to predict HLA alleles using multiple SNPs, based on observed linkage disequilibrium (LD) [59]. In Erlich's review of HLA typing (2012), he points out that this approach remains to be seen effectiveness for rare alleles and for applications where the ethnicity of samples is different from that of the training set [32]. It has been thought that the imputation-based method is unlikely to be used in a clinical setting requiring high resolution (i.e, four-digit resolution) [32], however it could be relatively reliable and cost effective alternative to actual HLA typing for population-scale studies.

2.3.2 Existing HLA Imputation Methods with Haplotype Information

LDMhc

To impute HLA types from multiple SNP markers, Leslie *et al.* (2008) used a hidden Markov model (HMM) based on approximate coalescent models to develop their LDMhc algorithm, with a leave-one-out cross-validation scheme for SNP selection [59]. Dilthey *et al.* (2010) subsequently developed integrated software HLA*IMP for imputing classical HLA alleles from SNP genotypes based on LDMhc, with a modified SNP selection function that leads to pronounced increases in call rate [28]. However, a training set of SNP haplotypes with known HLA alleles are required by LDMhc, as well as a fine genetic map of the region [28], whereas most experimental techniques for determining SNPs provide genotypes rather than haplotypes. Inferring haplotypes from genotypes can be done with the statistical method of approximate coalescent models, PHASE [102], or newer algorithms like fastPHASE [97], MACH [65], IMPUTE2 [48], BEAGLE [13], SHAPEIT2 [26] and HAPI-UR [115].

The hidden Markov model (HMM) has been described informally by the "product of approximate conditionals" (PAC) models proposed in Li and Stephens (2003) [61]. The method of LDMhc assumes that if the additional haplotype carries a given HLA allele, it will look like an imperfect mosaic of those training haplotypes that carry the same allele. The degree of mosaicism is determined by the recombination rate, mutation rate and the number of training haplotypes that carry the allele in the database [59]. Formally, I described technical details of the algroithm as follows.

The basic idea is to calculate the posterior probability of a specified HLA allele y

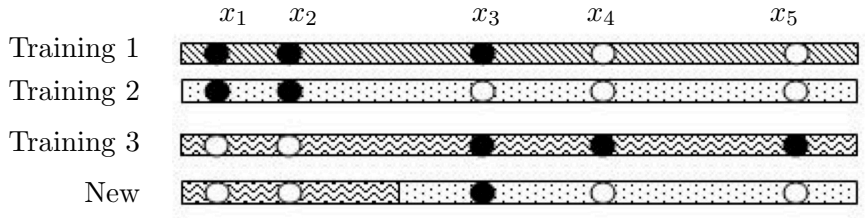


Figure 2.3: Illustration of an imperfect mosaic of a new haplotype from the training haplotypes carrying the same HLA allele. The hidden states, $(Z_1, \ldots, Z_5)$ for which training haplotype copies at the sites is (3, 3, 2, 2, 2), and there is a mutation at site 3.

given a new SNP haplotype $\{x_1^{\text{new}}, ..., x_m^{\text{new}}\}$. By Bayes rule,

$$\Pr(y \mid \{x_1^{\text{new}}, ..., x_m^{\text{new}}\}) = \frac{p_y^{\text{prior}} \pi(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y)}{\sum_{\forall y'} p_{y'}^{\text{prior}} \pi(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y')} , \tag{2.1}$$

assuming flat prior probability (i.e, $p_y^{\text{prior}} = p_{y'}^{\text{prior}}$ for $\forall\, y, y'$), where $\pi(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y)$ denotes the probability of the additional haplotype $\{x_1^{\text{new}}, ..., x_m^{\text{new}}\}$ given by a specified HLA allele y. The prediction is

$$\arg\max_y \hat{\Pr}(y \mid \{x_1^{\text{new}}, ..., x_m^{\text{new}}\}) \tag{2.2}$$

"Li and Stephens" uses an HMM formulation to approximate $\pi(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y)$, which allows relatively efficient computation [59]. Assume that there are n_y copies of HLA allele y in the training set, with the i^{th} training SNP haplotypes $\{x_{1,i}^{(y)}, ..., x_{m,i}^{(y)}\}$ among n_y where $1 \leq i \leq n_y$ and the superscript (y) indicates that the chromosome carries the HLA allele y. $\hat{\pi}(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y)$ builds $\{x_1^{\text{new}}, ..., x_m^{\text{new}}\}$ as an imperfect mosaic of $\{x_{1,i}^{(y)}, ..., x_{m,i}^{(y)}\}$ for all $1 \leq i \leq n_y$. That is, at each locus, x_j^{new} is a (possibly imperfect) copy of one of $\{x_{j,1}^{(y)}, ..., x_{j,n_y}^{(y)}\}$ at that position, taking mutation into account. The hidden states, $(Z_1, ..., Z_m)$ denote which of the training haplotype copies

at the loci $1, ..., m$ where $1 \leq Z_j \leq n_y$.

It is an illustration of an imperfect mosaic of $\{x_1^{\text{new}}, ..., x_m^{\text{new}}\}$ from $\{x_{1,i}^{(y)}, ..., x_{m,i}^{(y)}\}$, $1 \leq i \leq 3$ and $m = 5$, as shown in Figure 2.3. The first two white points at $\{x_1^{\text{new}}, ..., x_m^{\text{new}}\}$ are "inherited" from the third training haplotype $\{x_{1,3}^{(y)}, ..., x_{m,3}^{(y)}\}$, and the last three points are from the second one. However, the mutation effect is presented by a black point at site 3 of $\{x_1^{\text{new}}, ..., x_m^{\text{new}}\}$ instead of a white point from the second haplotype. Here, the hidden states $(Z_1, ..., Z_5) = (3, 3, 2, 2, 2)$.

In an HMM, we need to specify the start probability for hidden states, transition and emission probabilities. The start probability is defined as $\Pr(Z_1 = z) = 1/n_y$, where $z \in \{1, \ldots, n_y\}$. To mimic the effect of recombination of haplotype, Z_j is modeled as a Markov chain on $\{1, \ldots, n_y\}$, i.e. the distribution of Z_{j+1} only depends on the status of Z_j:

$$\Pr(Z_{j+1} = z_1 \mid Z_j = z_0) = \begin{cases} 1 - p_j + p_j/n_y & \text{if } z_1 = z_0 \\ p_j/n_y & \text{if } z_1 \neq z_0 \end{cases} \tag{2.3}$$

where $p_j = 1 - \exp\{-4N_e(r_{j+1} - r_j)/n_y\}$, N_e is the effective population size and assumed to be 15,000 in the model of Leslie *et al.* (2008) [59]; and $r_1, ..., r_m$ are genetic positions in Morgans, hence $r_{j+1} - r_j$ denotes the genetic distance between two adjacent loci.

Furthermore, the emission probability is expressed in terms of the population mutation rate θ_y for allele y:

$$\Pr(x_j^{\text{new}} = c \mid Z_j = z) = \begin{cases} \frac{n_y}{n_y + \theta_y} + \frac{1}{2}\frac{\theta_y}{n_y + \theta_y} & \text{if } x_{j,z}^{(y)} = c \\ \frac{1}{2}\frac{\theta_y}{n_y + \theta_y} & \text{if } x_{j,z}^{(y)} \neq c \end{cases} \tag{2.4}$$

where $j \in \{1, ..., m\}$. $\sum_{z=1}^{n_y-1} z^{-1}$ is the expected number of mutation events at a single site on the genealogical tree relating a random sample of n_y haplotypes [59], hence the population mutation rate is defined as $\theta_y = \left(\sum_{z=1}^{n_y-1} z^{-1}\right)^{-1}$.

Finally, we sum over all possible paths of Markov chain by using the standard forward algorithm of HMM, to calculate the conditional probability $\pi(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y)$.

$$f_{j+1}^y(t) = \Pr(x_{j+1}^{\text{new}} \mid Z_{j+1} = t) \sum_{z=1}^{n_y} f_j^y(z) \Pr(Z_{j+1} = t \mid Z_j = z) \tag{2.5}$$

where $f_1^y(t) = 1/n_y$, and then

$$\hat{\pi}(\{x_1^{\text{new}}, ..., x_m^{\text{new}}\} \mid y) = \sum_{t=1}^{n_y} f_m^y(t). \tag{2.6}$$

2.3.3 Existing HLA Imputation Methods Using Unphased Data

MAGprediction

A maximum likelihood method for obtaining sample haplotype frequencies under the assumption of random mating was suggested by Hill (1975) [44], using what has been recognized as an Expectation–Maximization (EM) algorithm [27], and this approach has been extended to a few loci [66, 34, 42]. Li et al. (2003) [63] proposed an approach based on the same likelihood formulation used in Excoffier and Slatkin (1995) [34], but adopting the estimating equation idea along with a modified progressive-ligation computational algorithm. Their method was implemented in a program HPlus. The primary purpose of HPlus method is to estimate haplotype frequencies with standard errors, and implicitly it could be applied to infer haplotype phase and genotype imputation. Later, Li et al. (2011) [64] extended HPlus method for the problem of HLA genotype imputation.

The basic idea of MAGprediction is to estimate the posterior probabilities of HLA type at the locus of interest given by a SNP genotype profile $\{g_1, ..., g_m\}$.

$$\Pr(\langle y^{(1)}, y^{(2)} \rangle \mid g_1, ..., g_m) \sim \Pr(\langle y^{(1)}, y^{(2)} \rangle, g_1, ..., g_m) \tag{2.7}$$

The joint probability over SNP genotypes and HLA type $\Pr(\langle y^{(1)}, y^{(2)} \rangle, g_1, ...g_m)$ can be estimated from training genotypes via haplotype frequencies. Under the HWE assumption the joint probability is the summation of frequency product of two associated haplotypes:

$$\Pr(\langle y^{(1)}, y^{(2)} \rangle, g_1, ...g_m) = \sum_{\Omega} f_{\{y', x'_1, ..., x'_m\}} f_{\{y'', x''_1, ..., x''_m\}} \tag{2.8}$$

where the set Ω represents all haplotype pairs of $\{y', x'_1, ..., x'_m\}$ and $\{y'', x''_1, ..., x''_m\}$ whose genotypes are consistent with the observed ones $\{\langle y^{(1)}, y^{(2)} \rangle, g_1, ..., g_m\}$, and $f_{\{y, x_1, ..., x_m\}}$ denotes the frequency of a haplotype $\{y, x_1, ..., x_m\}$.

Li *et al.* (2011) adopts an estimating equation approach, which has been used and detailed elsewhere (Li *et al.*, 2003, and Li *et al.*, 2007) [63, 62]. Briefly, they used the log likelihood function to derive its score estimating equation by the first derivative with respect to haplotype probabilities, an iterative procedure is conducted to find the score estimating equation estimate. To overcome the computational challenge, a modified progressive-ligation computational algorithm is used.

The haplotype frequencies $\hat{f}_{\{y, x_1, ..., x_m\}}$ are estimated from the training data, and lead to estimated conditional probabilities of genotypes

$$\hat{\Pr}(\langle y^{(1)}, y^{(2)} \rangle \mid g_1, ..., g_m) \tag{2.9}$$

Given by a new individual with SNP profile $\{g_1^{\text{new}}, ..., g_m^{\text{new}}\}$, the prediction is

$$\arg\max_{\langle y^{(1)}, y^{(2)} \rangle} \hat{\Pr}(\langle y^{(1)}, y^{(2)} \rangle \mid g_1^{\text{new}}, ..., g_m^{\text{new}}) \tag{2.10}$$

As being similar to the method proposed by Leslie et al. (2008), a small set of SNPs is selected before applying MAGprediction using a forward selection algorithm with the Akaki criterion (AIC) [64]. The author of MAGprediction claimed that the predictive model should have the fewest predictive SNPs possible, without sacrificing predictive accuracy.

BEAGLE

BEAGLE is based on a model that locally clusters haplotypes [13, 14]. The localized haplotype-cluster model depends on the fitting of variable-length Markov chain models, which automatically adapt to the degree of linkage disequilibrium between markers to create a parsimonious model for the LD structure [13].

In the model of BEAGLE, the observed haplotypes are grouped into clusters at each marker position, based on similarity of the haplotypes at markers in the local vicinity. As one moves along the model from one marker to the next, cluster membership tends to stay stable, with some changes due to historical recombination or mutation events. The localized haplotype-cluster model can be interpreted as a special class of HMMs for haploid data, and this model is extended to a diploid HMM for use of efficient HMM sampling algorithms on unphased data.

In the genetic association study of Raychaudhuri et al. (2012), BEAGLE was used to impute HLA alleles and all SNP across the HLA region were included simultaneously with HLA genes of interest in the imputation without variable deletion.

2.4 Proposed Method – HIBAG

We propose the HIBAG algorithm to impute HLA types, using the bagging method developed by Breiman [9, 10, 11], with improvements of variable subset suggested by random forests [12] and Bryll et al. [17], applied to a haplotype-based classifier. By randomly sampling sets of individuals from a training data set and randomly selecting SNPs from the available SNPs, we end up with an ensemble classifier that performs well in predicting HLA types. We describe how we develop a set of individual predictors and then how a user may apply these predictors in a particular case.

We begin with a set of individuals T that have both HLA alleles and SNPs genotyped in the xMHC and we take a series of K bootstrap samples (with replacement), B_k, of individuals from this set, $k = 1, 2, \ldots, K$. Each B_k is of size n, including some individuals from T that appear more than once and some that do not appear at all. Unselected samples form an "out-of-bag" set for the k^{th} selection. Breiman [11] pointed out that about $1/e \approx 37\%$ of T are out-of-bag for any B_k. We construct a classifier C_k for B_k that estimates HLA types using an optimal subset, $\mathcal{S}_k$, of the SNPs. In the following sections we describe construction of the classifiers C_k and selection of the SNP set $\mathcal{S}_k$.

2.4.1 Individual Classifiers

HLA and SNP genotypes available for individuals for each bootstrap sample B_k from T are used to form haplotypes and their estimated frequencies (Figure 2.1) using the EM algorithm assuming HWE [44] as extended to multiple loci [66, 92]: multi-locus genotype frequencies are assumed to be the sum of products of haplotype frequencies. Since the number of possible resolutions of phase increases exponentially with the number of heterozygous loci, a progressive ligation (PL) computational strat-

egy [80] is used, in which rare haplotypes with frequency less than 10^{-5} are ignored in order to achieve a computationally tractable algorithm.

The individual classifier C_k is built using the probability of all possible HLA types given the SNP profile observed at $\mathcal{S}_k$. The conditional probability follows from the joint probability of an HLA type and the SNP genotypes and this, in turn, is the sum, over all pairs of haplotypes that are consistent with the observed genotypes, of the products of frequencies of those two haplotypes. For example, HLA heterozygote A_1A_2 and one-locus SNP heterozygote profile s_1s_2 requires summation over two pairs of haplotypes (A_1s_1, A_2s_2) and (A_1s_2, A_2s_1). The pair of HLA alleles that maximizes this sum is the prediction of that individual classifier.

The posterior probabilities of HLA types given by the SNP genotypes at a set of loci $\mathcal{S}_k$ for C_k:

$$\Pr(\langle y^{(1)}, y^{(2)}\rangle \mid g_j, ...\forall j \in \mathcal{S}_k) \propto \Pr(\langle y^{(1)}, y^{(2)}\rangle, g_j, ...\forall j \in \mathcal{S}_k) \tag{2.11}$$

where K is the total number of classifiers.

The joint probability over SNP genotypes for an HLA type $\Pr(\langle y^{(1)}, y^{(2)}\rangle, g_j, ...\forall j \in \mathcal{S}_k)$ can be estimated from training genotypes via haplotype frequencies. Since unexpected haplotypes sometimes are observed in the genotypes of new individuals due to genotyping error, SNP mutation or rare haplotypes with $< 10^{-5}$ frequency, and then an error rate per locus (10^{-5}) is used. Under the HWE assumption the joint probability is the summation of frequencies of two associated haplotypes:

$$\Pr(\langle y^{(1)}, y^{(2)}\rangle, g_j, ...\forall j \in \mathcal{S}_k) = \sum_{\Omega} f_{\{y', x'_j, ...\forall j \in \mathcal{S}_k\}} f_{\{y'', x''_j, ...\forall j \in \mathcal{S}_k\}} \tag{2.12}$$

where the set Ω represents all haplotype pairs of $\{y', x'_j, ...\forall j \in \mathcal{S}_k\}$ and $\{y'', x''_j, ...\forall j \in$

$\mathcal{S}_k\}$ whose genotypes are consistent with the observed ones $\{\langle y^{(1)}, y^{(2)}\rangle, g_j, ...\forall j \in \mathcal{S}_k\}$, and $f_{\{y,x_j,...\forall j\in\mathcal{S}_k\}}$ denotes the frequency of a haplotype $\{y, x_j, ...\forall j \in \mathcal{S}_k\}$.

The haplotype frequencies $\hat{f}_{\{y,x_j,...\forall j\in\mathcal{S}_k\}}$ are estimated from the bootstrap sample B_k of training data T, and lead to estimated conditional probabilities of genotypes

$$\hat{\Pr}_k(\langle y^{(1)}, y^{(2)}\rangle \mid g_j, ...\forall j \in \mathcal{S}_k) \tag{2.13}$$

in Eqn (2.11). For a new individual with SNP profile $\{g_1^{\text{new}}, ..., g_m^{\text{new}}\}$, the prediction of C_k is

$$\underset{\langle y^{(1)},y^{(2)}\rangle}{\arg\max}\, \hat{\Pr}_k(\langle y^{(1)}, y^{(2)}\rangle \mid g_j^{\text{new}}, ...\forall j \in \mathcal{S}_k) \tag{2.14}$$

2.4.2 SNP Selection

In building each classifier, we select a subset $\mathcal{S}_k$ of SNPs for predicting HLA types and assure a computationally tractable method. The selection of $\mathcal{S}_k$ includes a random and a deterministic component, iteratively sampling a subset m_{try} of the m total SNPs at random, adding each of m_{try} SNPs to C_k one at a time, and adding the SNP that results in the highest out-of-bag prediction accuracy to $\mathcal{S}_k$. This process is repeated, adding one SNP at a time to $\mathcal{S}_k$, until no further improvement in prediction of HLA types is achieved by adding additional SNPs. The procedure does not guarantee the optimal subset to be found and there may be many sub-optimal SNP sets, then this algorithm just randomly pick one of them.

We set m_{try} to be much less than m (the total number of SNPs) to increase the independence of individual classifiers and reduce the variance of the ensemble by distributing classifiers semi-randomly over all SNPs. If m_{try} is too small compared to m the variable selection approach is likely to select less-informative SNP markers.

Box 2.4.1 The attribute bagging algorithm.

1. For $k = 1$ to K:
   - Draw a bootstrap sample B_k of size n (# of training samples) with replacement from the training data, and B_k contains all candidate SNPs.
   - Build an individual classifier C_k on B_k by variable selection with a random component (in Box 2.4.2), where C_k uses only a small set of SNPs $\mathcal{S}_k$.
2. Output the ensemble of individual classifiers $C = \{C_k\}_1^K$.
3. To predict the HLA type of a new individual with SNP profile $\{g_1^{\text{new}}, ..., g_m^{\text{new}}\}$, we average the posterior probabilities $\hat{\Pr}_k(\langle y^{(1)}, y^{(2)}\rangle \mid g_j^{\text{new}}, ...\forall j \in \mathcal{S}_k)$ among all C_k. The prediction is

$$\arg\max_{\langle y^{(1)}, y^{(2)}\rangle} \frac{1}{K} \sum_{k=1}^{K} \hat{\Pr}_k(\langle y^{(1)}, y^{(2)}\rangle \mid g_j^{\text{new}}, ...\forall j \in \mathcal{S}_k) \tag{2.15}$$

Although this would not necessarily reduce accuracy, it would require larger numbers of classifiers. In general, reducing m_{try} reduces both the correlation and the strength of individual classifiers, whereas increasing it increases both. We have found a value of $m_{\text{try}} = \sqrt{m}$ to perform well, as shown in Table 2.4. This rule is a recommendation of the random forest method (Hastie et al., 2009, Section 15.3) [41].

The attribute bagging algorithm is shown in Box 2.4.1 and Box 2.4.2. K is the total number of individual classifiers, and m_{try} is the number of variables randomly sampled as candidates for selection ($m_{\text{try}} = \sqrt{m}$ by default). For each step of adding a new predictor, m_{try} variables are re-drawn randomly from the candidate SNPs.

We find in our experiments that $K = 25$ is sufficient to give a highly accurate and stable ensemble classifier, and this number of bootstrap replicates was also observed by Breiman (1996) [9]. The prediction accuracy is not sensitive to m_{try} since all of the SNP markers are determined whether or not they are included in the SNP set for

Box 2.4.2 The algorithm of variable selection with a random component for an individual classifier C_k.

1. The set of loci for estimating haplotype frequencies is initially set to $\mathcal{S}_k = \{\emptyset\}$.
2. Build an individual classifier C_k on the bootstrapped data B_k, by recursively repeating the following steps from i) to iii) until it is not possible to reduce the losses[1]:
   - i) Select m_{try} SNP markers at random without replacement from the m total candidate SNPs ($m_{\text{try}} < m$) except the marker(s) in $\mathcal{S}_k$.
   - ii) Select the best SNP marker j^* based on the criteria of losses[1] among m_{try}.
   - iii) Add the selected SNP marker to the set $\mathcal{S}_k \leftarrow \mathcal{S}_k + \{j^*\}$.
3. Output C_k and $\mathcal{S}_k$.

[1]: reduce both the 0-1 loss and log likelihood loss of C_k (see **the details of loss criteria** in Sec 2.4.2).

each construction of individual classifier. The motivation for selecting $m_{\text{try}} < m$ is to increase the independence of individual classifiers and reduce the variance of the ensemble by distributing classifiers randomly over all SNPs. However, if m_{try} is too small compared to m (e.g., $m_{\text{try}} = 1$), the variable selection approach is likely to select non- or less- informative SNP markers. Although this would not necessarily reduce accuracy, it would require larger numbers of classifiers. In general, reducing m_{try} reduces both the correlation and the strength of individual classifiers, and increasing it increases both, therefore the optimal range of m_{try} could be usually quite wide. In our experiments the parameter setting $m_{\text{try}} = \sqrt{m}$ is appropriate for both a small set as well as hundreds of SNP predictors after taking the computational burden into account. In this study, we used the parameter settings $K = 100$, $m_{\text{try}} = \sqrt{m}$ for all tables and figures.

Details of the Loss Criteria:

Each bootstrap sample B_k leaves out $1/e \approx 37\%$ of the training samples, and these left out samples can be used to form accurate estimates (called out-of-bag estimation) [11]. The 0-1 loss of C_k is calculated from the out-of-bag samples to avoid over-fitting of that individual classifier. We minimize the 0-1 loss first, and if the 0-1 losses equal each other then choose the SNP marker with the lowest log likelihood loss. We add SNP markers until it is not possible to further reduce both the losses. Unlike the traditional variable selection with a penalty for the number of parameters, e.g., the Akaike criterion (AIC), our approach adds as many SNP predictors as possible to avoid variable searching stopping too early. Although more variables in an individual classifier result in greater computational complexity and larger variance of estimates, bagging and use of different variable subsets help to improve the stability of ensemble classifier [17, 9]. We therefore control model over-fitting twice, first at the level of the individual classifier and then at the level of aggregation in the ensemble.

The 0-1 loss of C_k is calculated from the out-of-bag samples to avoid over-fitting of that individual classifier,

$$\text{0-1 loss} = 1 - \text{accuracy of out-of-bag samples}$$

and the log likelihood loss of C_k is computed to assess fitting the model of haplotype frequencies with the assumption of HWE using the in-bag samples B_k,

$$\text{log likelihood loss} = -2 \times \sum_{i=1}^{n} \ln \hat{\Pr}_k(\langle y^{(1)}, y^{(2)} \rangle_i \mid g_{j,i}, ... \forall j \in \mathcal{S}_k)$$

where the subscript i indicates the i^{th} individual in B_k.

2.4.3 Bootstrap Aggregating

HIBAG is an ensemble classifer which employs bootstrap aggregation, known as bagging. The ensemble classifier is created from K bootstrap samples each using a set $\mathcal{S}_k$ of SNPs and to build a single classifier C_k. A comparison of accuracies among different model parameters m_{try} and K is shown in Table 2.4.

Application of HIBAG to a subject with the required SNP genotypes estimates the probability of each possible HLA types for all K classifiers. The process of aggregating (averaging over) the K predictors results in greater precision in the prediction probabilities. In this study we choose the HLA type with the highest probability averaged over the K probabilities as the final predicted genotype for estimating measures of prediction quality. However, in other applications, such as in the analysis of genotype-phenotype relationships, the vector of genotype probabilities may be preferred [70].

2.4.4 Implementation

We implemented the algorithm in an R package – HIBAG, which is available at R CRAN (`http://cran.r-project.org/web/packages/HIBAG/index.html`). To facilitate future use of this method, we have prepared pre-built classifiers based on our Study Data which can be used to impute HLA alleles in new SNP data, which are available at `http://www.biostat.washington.edu/~bsweir/HIBAG/`. These classifiers were constructed using the training data sets, as reported in this paper with supporting test results, and the combined training and testing data to yield the most accurate predictions. This enables users to apply the HIBAG method without needing access to a training data set. Alternatively, the software can build new classifiers from

Table 2.4: Assessing the prediction accuracies using different model parameter settings, when STUDY Data[1] of European ancestry were divided into training and validation sets with approximately equal sizes[2]. No call threshold was executed.

	HLA –						
Accuracy (%)	*A*	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
# of SNPs [3]	273	341	356	327	349	356	279
# of training samples	945	1314	944	1234	874	968	820
# of validation samples	912	1258	922	1202	866	956	804
The total number of classifiers $K = 25$							
$m_{\text{try}} = 1$	98.1	95.9	98.3	91.4	96.4	97.4	93.3
$m_{\text{try}} = \sqrt{m}$	98.4	96.4	98.5	92.0	97.2	98.6	93.7
$m_{\text{try}} = \frac{1}{3}m$	98.2	95.7	99.0	91.3	96.7	98.6	93.5
$m_{\text{try}} = m$	98.5	95.7	99.0	91.5	96.7	98.5	93.5
The total number of classifiers $K = 100$							
$m_{\text{try}} = 1$	98.2	96.1	98.3	92.1	96.6	97.7	93.4
$m_{\text{try}} = \sqrt{m}$	98.2	96.6	98.8	92.1	97.3	98.8	93.8
$m_{\text{try}} = \frac{1}{3}m$	98.4	95.8	99.1	91.5	96.8	98.7	93.7
$m_{\text{try}} = m$	98.5	95.8	99.0	91.7	96.7	98.7	93.4

[1]: STUDY Data refers to Section 2.5.4.
[2]: K is the total number of individual classifiers, m is the total number of SNP markers, and m_{try} is the number of variables randomly sampled as candidates for each selection.
[3]: SNP markers common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms within a flanking region of 500kb are used.

training data supplied by the user, and HIBAG is computationally feasible for much large training samples. As the construction of individual classifiers are independent from each other, building an ensemble model in parallel is possible. As an example, it takes about 52 minutes to build an individual classifier of *HLA–A* on the training samples of European ancestry data ($n = 1504$) with 273 SNP markers on average. More details are shown in Table 2.5. The computation time while using the published parameters is much less since no training is needed, e.g., the algorithm takes at most 41 minutes for predicting 100 new individuals at *HLA–B* locus, since no training is needed.

Table 2.5: Assessing the computational times (hour) of building a HIBAG model for our published parameter estimates of European ancestry on a Linux system with Intel processor (2.27GHz) and 32 GB RAM.

	HLA –						
	A	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
# of SNPs [1]	273	341	356	327	349	356	279
# of HLA alleles	48	88	37	55	17	21	26
# of training samples	1504	2030	1493	1909	1380	1517	1274
Building a HIBAG model: computing time per individual classifier							
	0.86h	6.12h	0.84h	3.36h	0.58h	0.56h	0.28h

[1]: SNP markers common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms within a flanking region of 500kb are used.

2.5 Materials

The numbers of individuals with available four-digit HLA types and the numbers of observed HLA alleles are summarized in Table 2.6 for the HapMap, WTCCC and HLARES data respectively. Note that sample sizes vary among HLA loci due to missing data. Descriptions of these data follow.

2.5.1 HapMap Data

The HapMap Phase 2 SNP dataset consists of 1) 30 parent-offspring trios of Yoruban ancestry from Ibadan in Nigeria, YRI; 2) 30 CEPH trios of European ancestry from Utah, CEU; 3) 45 unrelated Han Chinese from Beijing, CHB; 4) 45 unrelated individuals from Tokyo in Japan, JPT. The HapMap SNP genotypes (release #28) were downloaded from `http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/2010-08_phaseII+III/forward/`. The dataset was created by combining genotyping data from several platforms: Affymetrix, Illumina, Perlegen, etc. When Mendelian errors were detected in a trio, all genotypes for that SNP in that trio were set to missing.

Table 2.6: The numbers of individuals with four-digit HLA types and the observed number of HLA alleles for each locus.

	HLA –						
	A	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
Individuals genotyped							
HapMap							
CEU	90	68	90	90	90	90	90
YRI	90	88	88	89	90	88	12
CHB+JPT	89	89	89	88	89	87	58
WTCCC							
European	884	1532	840	1129	0	1004	0
HLARES							
European	1857	2572	1866	2436	1740	1924	1624
Asian	517	624	522	608	495	525	469
Hispanic	298	430	300	420	269	312	263
African	80	112	80	102	74	78	69
Unique HLA alleles							
European	48	88	37	55	17	21	26
Asian	43	72	34	49	17	19	29
Hispanic	41	85	31	44	14	17	26
African	36	45	24	30	13	17	23
Total	85	144	49	80	19	27	49

SNP markers were selected within the extended MHC (xMHC) [47] on chromosome 6 ranging from 025759242bp to 033534827bp. With a missing call rate threshold of 10%, there were 16241, 17160, and 16896 SNP markers in the xMHC for CEU, YRI and CHB+JPT respectively.

High resolution classical HLA data for *HLA–A*, *B*, *C*, *DRB1*, *DQA1*, *DQB1* and *DPB1* were derived by combining genotypes previously published for these samples [25] with sequence base typing (SBT) data generated by Conexio Genomics (Perth, Australia).

2.5.2 WTCCC Data

SNP and HLA genotypes for the British 1958 birth cohort (`http://www.b58cgene.sgul.ac.uk/`) were downloaded from the European Genotype Archive (EGA, `http://www.ebi.ac.uk/ega/`). Candidate SNP markers from Illumina Human1M-Duo platform [114] were selected within the xMHC with a 10% threshold of missing SNP genotypes. The final data set included 2,922 unrelated individuals and 7,601 SNP markers. The HLA data description is available at `https://www-gene.cimr.cam.ac.uk/todd/public_data/HLA/HLA.shtml`. Five HLA loci, *HLA–A*, *B*, *C*, *DRB1* and *DQB1*, were typed to four digits using Sequence Specific Oligonucleotide (SSO) and Sequence Specific Primer (SSP) methodologies.

2.5.3 HLARES Data

SNP data from the xMHC typed using the Illumina 1M and 1M Duo platforms and HLA data were aggregated from several GlaxoSmithKline clinical trials, including subjects of European ($n = 2,668$), Asian ($n = 720$), Hispanic ($n = 439$) and African ($n = 173$) ancestries. There were 7,976 xMHC SNP markers available with less than 10% missing data. HLA data for GSK clinical trial samples were generated by Conexio Genomics (Perth, Australia), HistoGenetics (NY, USA) and LabCorp (NC, USA) using SBT, SSO and SSP methodologies for *HLA–A*, *B*, *C*, *DRB1*, *DQA1*, *DQB1* and *DPB1*. The smallest sample size of African ancestry among four ethnic groups led to fewest unique HLA alleles observed, although populations of African ancestry would be expected to have highest genetic diversity.

2.5.4 Data for Performance Assessment

To assess HIBAG performance and build broadly applicable classifiers included in the HIBAG R package, a set of 1,564 SNP markers within the xMHC were selected that were available in all three samples and common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms. In the sensitivity analysis, 5,316 SNPs within the xMHC genotyped on Illumina 1M Duo platform were used.

The individuals of this study were self-reported as either European, Asian, Hispanic or African descent. HLA and SNP genotypes of performance assessment (here referred to as "STUDY Data") consist of 1) HLARES data of European ancestry, 2) HLARES data of Asian ancestry and HapMap CHB+JPT, 3) HLARES data of Hispanic ancestry, and 4) HLARES data of African ancestry and 60 parents of HapMap YRI.

2.6 Evaluation of HIBAG

We evaluated the performance of HIBAG by building the classifier using a training sample and imputing HLA types in an independent testing sample, and compared the imputed genotypes to experimentally determined HLA types. As further evaluation, we compared the performance of HIBAG with BEAGLE and HLA*IMP.

2.6.1 Measures of Prediction Quality

Prediction accuracy was used to assess overall model performance, defined as "the number of chromosomes with HLA alleles predicted correctly" over "the total number of chromosomes". In addition, sensitivity (SEN), specificity (SPE), positive predictive value (PPV) and negative predictive value (NPV) were used to evaluate the predictive performance for each HLA allele. These standard statistical quantities are

		True HLA allele		
		H	Not *H*	
Prediction	*H*	True Positive	False Positive	**Positive Predictive Value** = # of "true positive" / # of "prediction is *H*"
	Not *H*	False Negative	True Negative	**Negative Predictive Value** = # of "true negative" / # of "prediction is not *H*"
		Sensitivity = # of "true positive" / # of "true allele is *H*"	**Specificity** = # of "true negative" / # of "true allele is not *H*"	**Allele Accuracy** = # of "true pos" + "true neg" / grand sum

Figure 2.4: The standard statistical quantities of prediction quality for a specific HLA allele H: sensitivity, specificity, positive predictive value, negative predictive value and allele accuracy.

defined in Figure 2.4. HIBAG produces a posterior probability for each possible HLA type. Placing a minimum threshold on the posterior genotype probability can increase prediction accuracy at the expense of reducing call rates. "Call" and "No Call" were determined by whether the posterior probability is greater or less than a call threshold.

2.6.2 Accuracy of Imputed HLA Types

We compared imputed to experimentally determined HLA types for four ancestries: European, Asian, Hispanic and African. For each ethnicity, we divided the STUDY Data into equal sized training and validation datasets. We built the HIBAG models using the training data and assessed the imputation accuracy with the independent testing data. We used the set of 1,564 MHC SNPs in common among several Illumina platforms for this analysis. The efficiency of the HIBAG algorithm was improved by restricting the number of possible SNPs that could be included in the model. We evaluated flanking regions from 50 to 1,000 kb to identify an appropriate

size for predicting HLA alleles. In subjects of European ancestry, the average accuracies reach their maximum values by 250kb (Figure 2.5). We conservatively chose to use a 500kb flanking region, including 1,042 SNPs, for subsequent imputation.

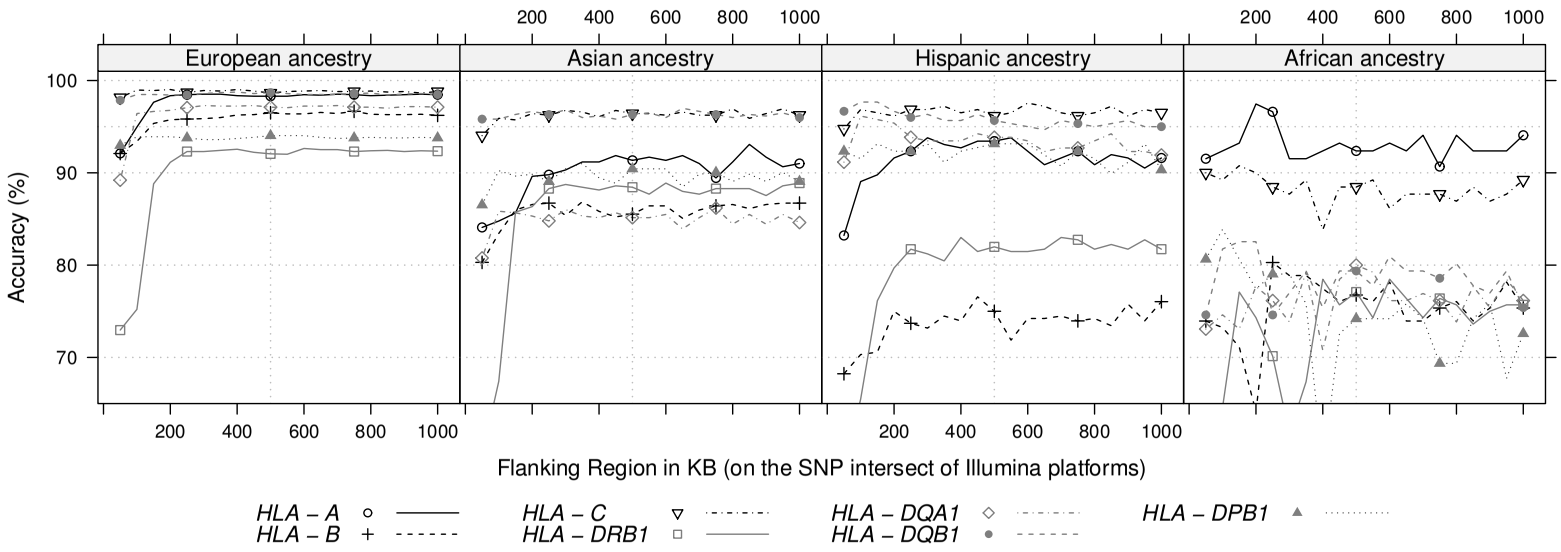


Figure 2.5: The relationships between the four-digit accuracies (no call threshold) and size of flanking region from 50kb to 1000kb on each side, stratified by HLA loci and ethnicities. SNP markers were genotyped on the intersect of Illumina platforms. A 500kb flanking region is an appropriate region for predicting HLA alleles.

We next investigated the influence that setting call thresholds on posterior probabilities has on calling accuracy and the trade-off that imposes on call rates. Using 500 kb of flanking markers around each HLA locus, $m_{\text{try}} = \sqrt{m}$ as the number of markers randomly sampled in building each classifier, and $K = 100$ bootstrap samples, we built the HIBAG models with European, Asian, Hispanic, and African ancestry training data sets respectively. The locus-specific calling accuracies were estimated from independent testing data sets of equal size as the training sets (Table 2.7). In Europeans, where we have the largest sample size, without any call threshold (CT = 0) the accuracies range from 92.1% to 98.8%. As observed previously (Leslie et al., 2008), *HLA–C* and *DQB1* yielded the highest prediction accuracies, closely followed by *A*, *DQA1* and *B*. The lowest accuracies were observed for *DPB1* and *DRB1*. Amongst non-Europeans, per locus accuracies were uniformly lower than in Europeans, and varied substantially from locus to locus. On average, the prediction accuracy was the lowest in subjects of African ancestry. These patterns are due to the differences in training sample size and several aspects of allelic heterogeneity, including number of alleles, their frequency distribution, and the degree of haplotypic mosaicism within four-digit alleles [59]. The results using all Illumina 1M MHC markers were not noticeably better than the intersection across several commonly used Illumina genome-wide panels as shown in Table 2.8. We therefore focused on the intersection as a more broadly applicable panel.

The prediction accuracy can be improved by taking the HIBAG posterior genotype probabilities into account, as those with higher probabilities have a higher likelihood of being a correct call. An empirical relationship between posterior probability and overall accuracy is shown in Figure 2.6. The improvement in accuracy comes at a cost of lower genotype call rates, as illustrated in Figures 2.7 and 2.8. All seven loci

Table 2.7: Summary of the four-digit prediction accuracies (call rates) stratified by ancestries and HLA loci. STUDY Data were divided into training and validation sets with approximately equal sizes. HIBAG call thresholds (CT) of 0 and 0.5 were used.

	HLA –						
	A	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
European ancestry							
# of SNPs [1]	273	341	356	327	349	356	279
HIBAG							
CT = 0.0	98.2(100)	96.6(100)	98.8(100)	92.1(100)	97.3(100)	98.8(100)	93.8(100)
CT = 0.5	98.7(98.8)	97.8(94.2)	99.2(98.0)	94.9(90.1)	97.8(97.9)	99.2(97.9)	94.8(96.0)
BEAGLE [2]							
	98.1(100)	95.5(100)	97.7(100)	92.9(100)	96.4(100)	97.9(100)	94.7(100)
Asian ancestry							
# of SNPs [1]	259	334	346	319	341	348	272
HIBAG							
CT = 0.0	92.1(100)	87.5(100)	96.6(100)	88.7(100)	86.8(100)	96.0(100)	89.8(100)
CT = 0.5	93.8(91.7)	94.7(71.0)	97.8(93.9)	95.8(71.5)	90.0(80.8)	98.1(96.3)	95.3(82.8)
BEAGLE [2]							
	93.8(100)	83.7(100)	94.5(100)	87.7(100)	86.7(100)	97.3(100)	91.2(100)
Hispanic ancestry							
# of SNPs [1]	274	341	356	326	348	355	278
HIBAG							
CT = 0.0	93.4(100)	75.0(100)	96.2(100)	82.0(100)	93.8(100)	95.7(100)	93.1(100)
CT = 0.5	96.0(82.5)	93.8(37.5)	98.4(87.4)	93.5(50.8)	95.8(90.8)	98.9(90.0)	97.5(81.5)
BEAGLE [2]							
	89.1(100)	75.0(100)	92.3(100)	78.7(100)	94.6(100)	96.3(100)	91.9(100)
African ancestry							
# of SNPs [1]	266	335	349	325	343	351	269
HIBAG							
CT = 0.0	92.4(100)	76.8(100)	88.5(100)	77.1(100)	80.0(100)	79.4(100)	74.2(100)
CT = 0.5	100(74.6)	96.7(21.1)	96.5(66.2)	100(22.2)	97.2(27.7)	97.7(34.9)	75.0(12.9)
BEAGLE [2]							
	93.2(100)	71.1(100)	86.9(100)	81.2(100)	79.2(100)	76.2(100)	79.0(100)

[1]: SNP markers common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms within a flanking region of 500kb are used.
[2]: No call threshold.

can achieve greater than 99% calling accuracy with sufficiently stringent choices of posterior probabilities; however, this would lead to call rates below 60% in the case of *DRB1* and *DPB1*. The best choice of call threshold for each locus will vary based on study criteria. We have selected a threshold of 0.5 as a value that has modest effects

Table 2.8: Summary of the four-digit accuracies from BEAGLE and HIBAG using the same SNP sets. Study data were randomly divided into training and validation sets with approximately equal sizes for each ancestry. No call threshold was used, and the SNP markers within a 500kb flanking region on each side were used.

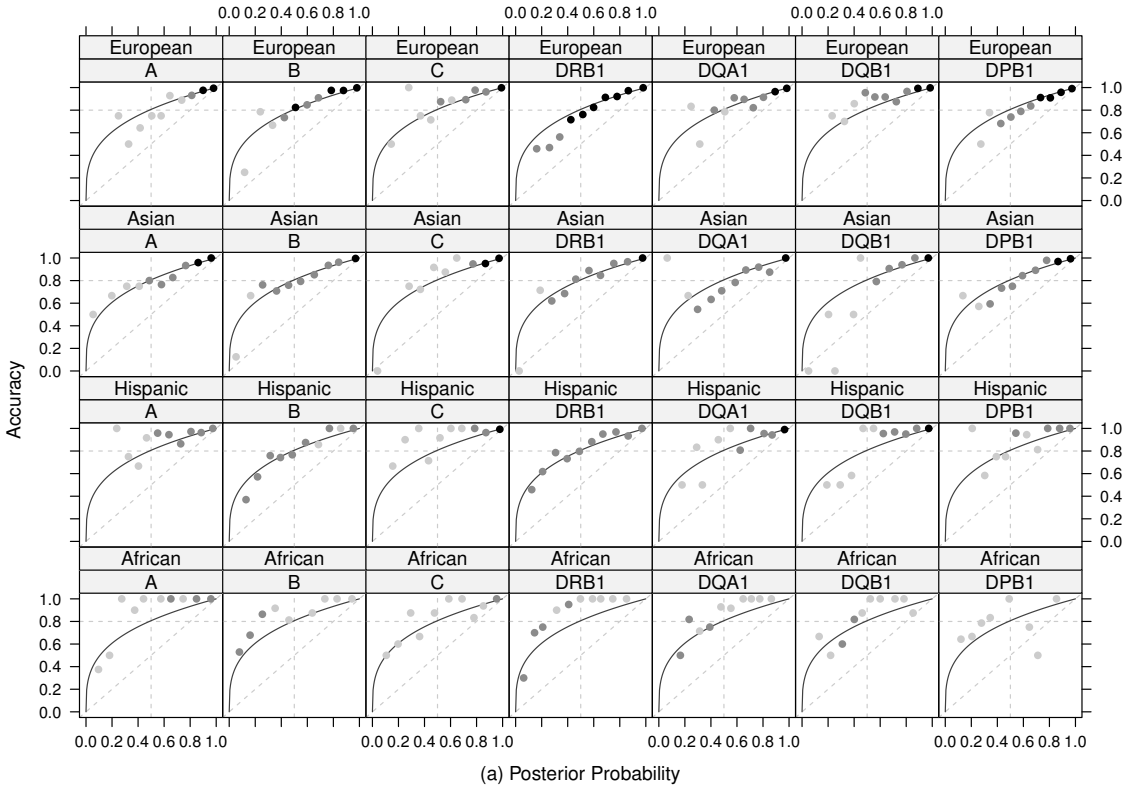
	HLA –						
Accuracy (%)	*A*	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
European ancestry							
# of training samples	945	1314	944	1234	874	968	820
# of validation samples	912	1258	922	1202	866	956	804
# of HLA alleles	48	88	37	55	17	21	26
# of SNPs[1] (1M/intersect)	937/273	942/341	979/356	921/327	964/349	979/356	786/279
BEAGLE – 1M	98.5	95.9	98.4	93.3	97.1	98.7	95.2
HIBAG – 1M	98.5	96.4	98.6	92.4	97.3	98.7	95.9
BEAGLE – Common	98.1	95.5	97.7	92.9	96.4	97.9	94.7
HIBAG – Common	98.2	96.6	98.8	92.1	97.3	98.8	93.8
Asian ancestry							
# of training samples	317	378	318	363	298	313	271
# of validation samples	289	335	293	333	286	299	256
# of HLA alleles	42	72	34	48	17	18	27
# of SNPs[1] (1M/intersect)	942/259	942/334	974/346	934/319	973/341	995/348	803/272
BEAGLE – 1M	93.4	87.0	96.6	89.5	87.0	98.2	91.8
HIBAG – 1M	92.1	87.8	96.9	91.7	89.2	98.2	91.0
BEAGLE – Common	93.8	83.7	94.5	87.7	86.7	97.3	91.2
HIBAG – Common	92.1	87.5	96.6	88.7	86.8	96.0	89.8
Hispanic ancestry							
# of training samples	161	238	157	223	139	162	139
# of validation samples	137	192	143	197	130	150	124
# of HLA alleles	41	85	32	44	14	17	26
# of SNPs[1] (1M/intersect)	965/274	966/341	996/356	954/326	992/348	1013/355	824/278
BEAGLE – 1M	88.7	75.8	92.0	78.4	94.2	97.7	94.9
HIBAG – 1M	91.6	74.0	95.5	82.0	96.9	98.0	95.6
BEAGLE – Common	89.1	75.0	92.3	78.7	94.6	96.3	91.9
HIBAG – Common	93.4	75.0	96.2	82.0	93.8	95.7	93.1
African ancestry							
# of training samples	81	100	74	89	69	74	44
# of validation samples	59	71	65	72	65	63	31
# of HLA alleles	36	45	24	30	13	17	23
# of SNPs[1] (1M/intersect)	949/266	948/335	981/349	945/325	983/343	1004/351	816/269
BEAGLE – 1M	96.6	70.4	87.7	73.6	86.2	84.1	85.5
HIBAG – 1M	95.8	81.0	90.0	84.0	85.4	80.2	82.3
BEAGLE – Common	93.2	71.1	86.9	81.2	79.2	76.2	79.0
HIBAG – Common	92.4	76.8	88.5	77.1	80.0	79.4	74.2

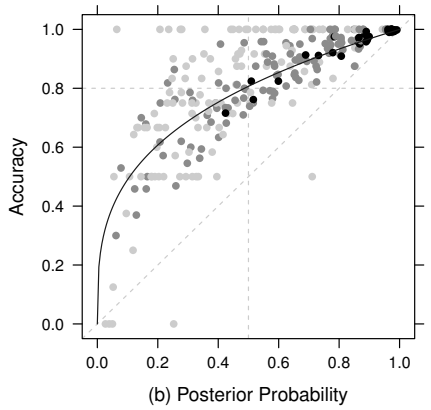
[1]: Illumina Human1M / Common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms.

on both call rate and accuracy. At this threshold, range in European accuracies increase to 94.8% to 99.2% with call rates between 90.1% and 98.8%. Among the non-Europeans, in some instances this threshold led to dramatic improvements in accuracies with corresponding decreases in call rates. For example, the accuracy

of *HLA–B* types in subjects of African ancestry improved from 76.8% to 96.7%, but with a call rate of only 21.1%. This highlights the importance of careful call threshold selection.

The performance summaries by HLA locus presented above are an average of the accuracies of each of the alleles observed in the testing data set, weighted by their corresponding frequencies. Details of the predictive characteristics of each HLA allele using a call threshold of 0.5 are summarized in Supplementary Table S1.2 – S1.5. Some alleles have very high accuracies while others are much lower. Alleles with low accuracy tend to have lower frequencies, as illustrated in Figure 2.9. Our study confirms that having ten copies of an allele in the database is generally sufficient to provide high sensitivity (greater than 90% except for *HLA–B* and *DRB1*) (Leslie et al., 2008). We found that in most instances where alleles are miscalled, there is one particular allele that is substituted for the correct one (Supplementary Table S1.2). For example, *HLA–DRB1*01:01* has an 8% allele frequency in Europeans and is miscalled just over 5.6% of the time. In every instance that *DRB1*01:01* is miscalled, it is called as *DRB1*01:02*. This miscall is reasonable since *DRB1*01:01* and *DRB1*01:02* both belong to the same serological antigen carried by an allotype *DRB1*01*.





- ≥ 50 individuals
- ≥ 10 and < 50 individuals
- < 10 individuals

the curve is fitted by a function $y = x^r$ with a parameter $r = 0.31$, and 0.5 posterior probability approximately corresponds to the prediction accuracy 80%.

Figure 2.6: The relationship between posterior probability and overall accuracy. The accuracies are calculated from ten bins of posterior probabilities: (a) stratified by HLA loci and ancestries; (b) over all HLA loci and ancestries.

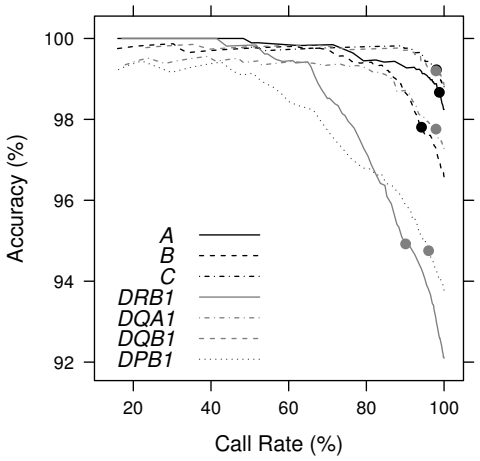


Figure 2.7: The relationship between accuracy and call rate when study data for individuals of European ancestry are divided into training and validation sets with approximately equal sizes. On the curve for each HLA locus, the 0.5 call threshold is indicated by •.

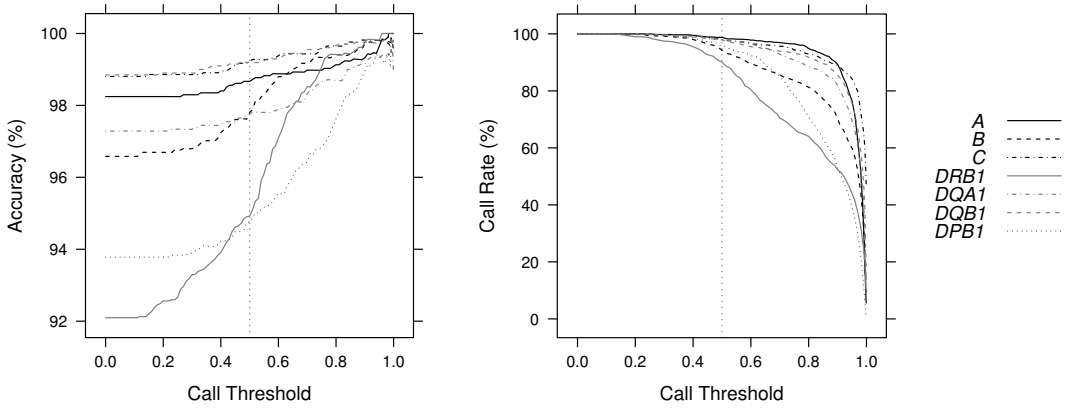


Figure 2.8: The relationships among call threshold, accuracy and call rate when study data for individuals of European ancestry are divided into training and validation sets with approximately equal sizes.

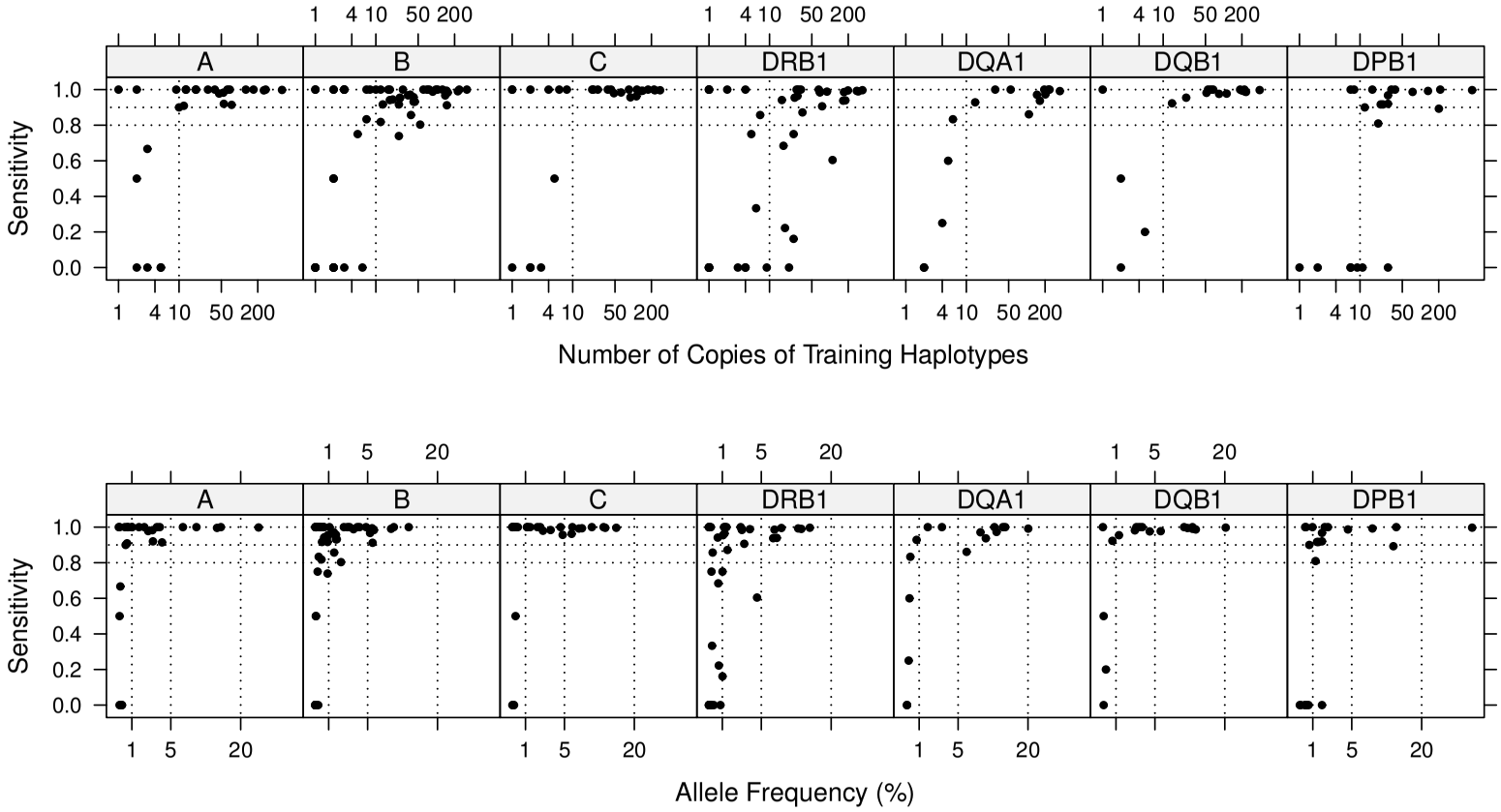


Figure 2.9: The relationship between four-digit sensitivities (no call threshold) and the number of copies of training haplotypes for each HLA allele when study data for individuals of European ancestry were divided into training and validation sets with approximately equal sizes. SNP markers on the intersect of Illumina platforms were used. For *A, C, DQA1, DQB1* and *DPB1*, 10 copies of training haplotypes seem sufficient to attain 90% sensitivity, but *B* and *DRB1* require many more training haplotypes.

2.6.3 Comparison with BEAGLE

BEAGLE is commonly used for genotype imputation and is unique among commonly used methods by accommodating multi-allelic variants [15]. It has been used to impute HLA types [93]. We therefore compared the performance of HIBAG to BEAGLE v3.3 (Tables 2.7 and 2.8). The default settings for BEAGLE were used, except that we increased the number of iterations from 10 to 50, which improve prediction accuracies. Since BEAGLE does not provide posterior probabilities for predicted HLA types, we compared BEAGLE's imputed HLA types to HIBAG's HLA types assuming no call threshold.

As shown in Table 2.7, the prediction accuracies of HIBAG and BEAGLE are similar. For samples of European ancestry, BEAGLE yields higher prediction accuracies than HIBAG at *HLA–DRB1* and *DPB1* (92.9% versus 92.1% and 94.7% versus 93.8%). However, HIBAG performed better at all other loci. For the non-European ancestries, the accuracies of BEAGLE and HIBAG are similar. A clear advantage of HIBAG over BEAGLE in the context of imputing HLA types is that HIBAG can be run efficiently using published classifiers whereas BEAGLE requires a training dataset.

*2.6.4 Comparison between HIBAG and HLA*IMP*

We employed the Oxford HLA imputation framework, HLA*IMP, to predict HLA alleles for STUDY Data with validation samples of European ancestry. The HLA*IMP method is implemented in a web based application with access to a training data set consisting of HapMap 30 CEU trios, 1958 birth cohort data of WTCCC, and a small number of additional samples from other projects [28]. Furthermore, using the

Table 2.9: The comparison of four-digit accuracies for HIBAG and HLA*IMP on the STUDY Data with no call threshold.

	HLA –				
Method	*A*	*B*	*C*	*DRB1*	*DQB1*
# of validation samples	1787	2471	1830	2383	1917
*Using 191 markers on Illumina 1M platform as selected by HLA*IMP* [1]					
# of SNPs	50	39	27	50	34
HLA*IMP (%)	91.0	94.4	98.4	87.9	96.2
HIBAG [2] (%)	96.7	94.8	98.7	90.0	98.6
Using all the xMHC markers on Illumina 1M platform [3]					
# of SNPs	489	570	560	476	448
HIBAG [2] (%)	97.7	95.2	98.7	91.8	98.4

[1]: The full SNP list is shown in Table S1.1.
[2]: The training samples are HapMap 30 CEU trios plus WTCCC samples.
[3]: The SNP markers within a flanking region of 250kb are used.

Illumina 1M option for HLA*IMP, we were able to identify SNPs used for prediction (Table S1.1). To enable a fair comparison, the training data set for HIBAG was limited to the HapMap 30 CEU trios and 1958 birth cohort data with the 191 SNPs selected by HLA*IMP for prediction. To illustrate the advantages of utilizing additional SNPs, we provide accuracy results for all Illumina 1M SNPs using the same training subjects. Results are summarized in Table 2.9. Since the call threshold for HLA*IMP refers to the posterior probability of an HLA allele rather than HLA type, the call rates correspond to different call thresholds. Figure 2.10 compares the performance of the three approaches by illustrating the relationship between four-digit prediction accuracy and call rate as shown in Figure 2.7. On the same set of 191 SNPs, HIBAG outperformed HLA*IMP at each locus, especially for *HLA–A* (accuracy = 96.7% versus 91.0%, respectively). As expected, using more SNP predictors yielded more accurate predictions although the gains were fairly modest.

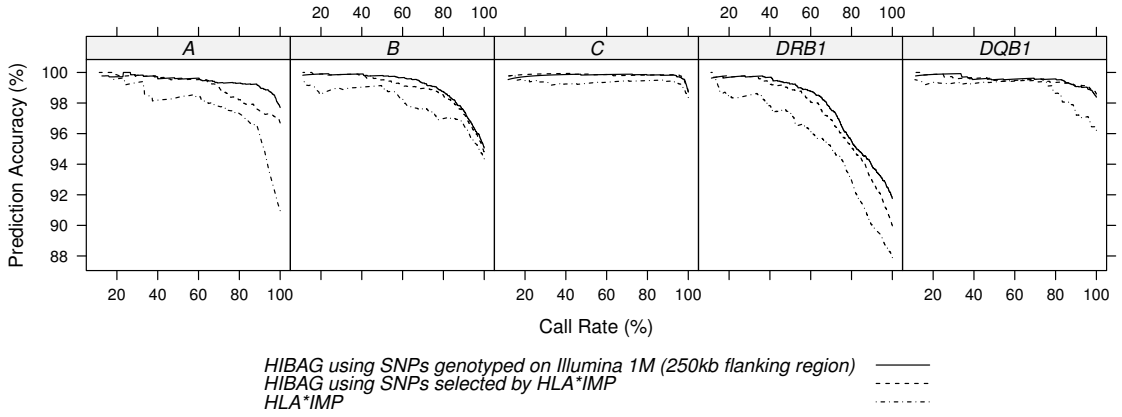


Figure 2.10: The relationship between four-digit accuracy and call rate for HLA*IMP and HIBAG respectively. Call rate is determined by the posterior probabilities of HLA allele (type) from HLA*IMP and HIBAG with a call threshold, and the accuracies are calculated from the validation samples with a call threshold.

Table 2.10: Summary of the four-digit prediction accuracies (call rates) for HLARES of European ancestry, using four-digit HLA data from the British 1958 birth cohort study as validation samples. HIBAG call threshold (CT) of 0 and 0.5 were used.

	HLA –				
	A	*B*	*C*	*DRB1*	*DQB1*
# of SNPs [1]	273	341	356	327	356
# of HLA alleles	48	88	37	55	21
# of training samples	1857	2572	1866	2436	1924
# of validation samples	884	1532	840	1129	1004
HLARES training data of European ancestry, the published pre-fit classifiers:					
CT = 0	98.1(100)	96.9(100)	96.5(100)	92.2(100)	97.8(100)
CT = 0.5	98.2(99.4)	97.4(97.3)	96.6(99.5)	94.0(94.6)	98.0(99.0)

[1]: SNP markers common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms within a flanking region of 500kb were used.

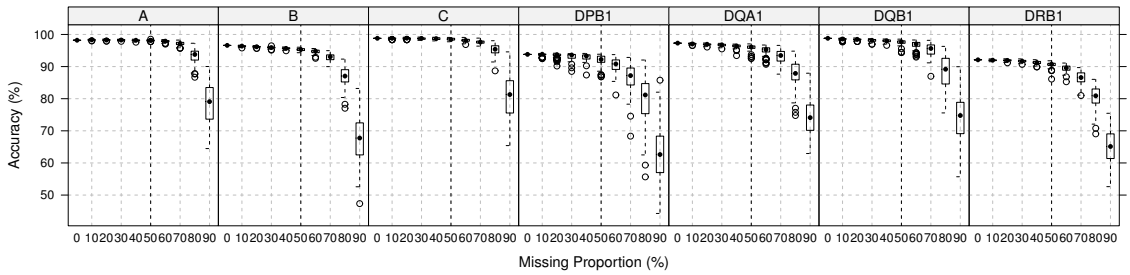
2.6.5 Miscellaneous

A practical issue is that HLA typing itself is not uniform between labs, therefore it is essential to evaluate the imputation method using data from different labs. A

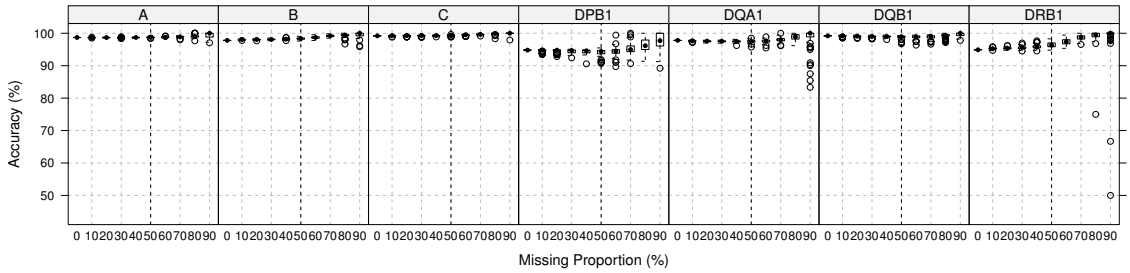
validation study was conducted to apply the published pre-fit classifiers to the data of the British 1958 birth cohort study. There are five HLA genes (*A*, *B*, *C*, *DRB1* and *DQB1*) available in this data. Table 2.10 summarizes the prediction accuracies of five HLA genes, ranging from 92% to 98%. *DRB1* is the most difficult to be predicted, and it is the second most polymorphic gene following *HLA–B*. These results are consistent with what Table 2.7 presents.

A simulation study indicates that the HIBAG method is robust to missing SNP markers with a fraction up to 50%, as shown in Figure 2.11. The missing SNP fraction of the original validation set is very small ($< 0.1\%$). For each run of simulation, we randomly remove a fraction of the SNP predictors used in the ensemble classifier (e.g, 10%, 20%) for the validation set where every validation sample has the same missing SNPs, and repeat this procedure 100 times. The box plots of accuracies (CT=0 and 0.5) and call rate are shown. The missing SNPs do not significantly reduce the accuracies for missing fractions less than 80%, but it does decrease the call rates. Furthermore, we also studied whether our method is sensitive to population structure or not, since HIBAG does assume Hardy-Weinberg equilibrium for calculating genotype probabilities. For each ethnicity, STUDY Data were divided into training and validation sets with approximately equal sizes as described in the previous section. A multi-ethnic HIBAG model was built using all training samples from multiple ethnicities, and then the accuracies were calculated for each validation set respectively. The result in Table 2.11 indicates that our method is robust to departures from HWE.

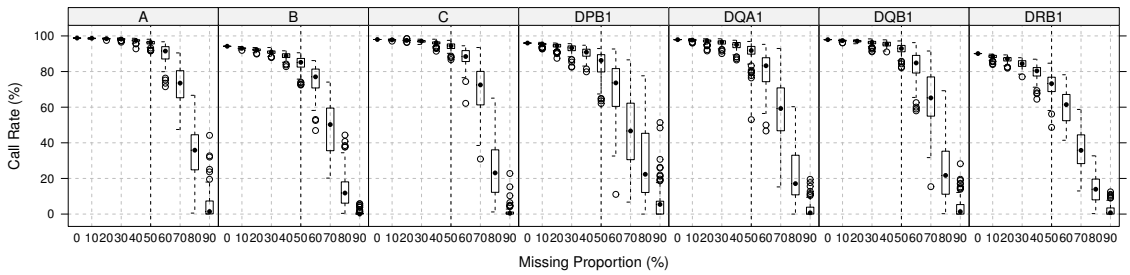
In our published pre-fit classifiers, 1042 SNPs were selected in total. With respect to the algorithm of HIBAG, variable selection is implicitly incorporated during the construction of each individual classifier, and more important SNP markers tend to be used more frequently in the ensemble. This use of selection frequency for SNPs



(a) no call threshold: accuracy vs. missing proportion.



(b) 0.5 call threshold: accuracy vs. missing proportion.



(c) 0.5 call threshold: call rate vs. missing proportion.

Figure 2.11: Box plots of accuracy and call rate with missing SNPs. Study data of European ancestry were divided into training and validation sets with approximately equal sizes. The HIBAG models were built using the training parts. For each run of simulation, a fraction of the SNP predictors used in the ensemble classifier (e.g, 10%, 20%) was removed randomly for the validation set, where every validation sample has the same missing SNPs, and repeat it 100 times. The missing SNPs do not significantly reduce the accuracies for missing fraction $< 80\%$, but it does decrease the call rates.

provides information to identify a small set of SNPs. The use frequencies of SNPs in the published pre-fit classifiers are shown in Figure 2.12. SNPs with low importance

Table 2.11: The accuracies calculated from the ethnic-specific and multi-ethnic models. For each ethnicity, STUDY Data were divided into training and validation sets with approximately equal sizes. The multi-ethnic model was built using all training samples from multiple ethnicities, whereas the ethnic-specific models were built using the training part of each ethnicity respectively. No call threshold was executed.

	HLA –						
	A	*B*	*C*	*DRB1*	*DQA1*	*DQB1*	*DPB1*
European ancestry							
multi-ethnic model	98.5	96.5	99.1	92.8	97.2	98.7	93.9
ethnic-specific model	98.2	96.6	98.8	92.1	97.3	98.8	93.8
Asian ancestry							
multi-ethnic model	89.1	85.8	95.6	87.2	86.2	96.5	90.4
ethnic-specific model	92.1	87.5	96.6	88.7	86.8	96.0	89.8
Hispanic ancestry							
multi-ethnic model	93.1	77.1	94.4	81.5	96.5	97.7	95.2
ethnic-specific model	93.4	75.0	96.2	82.0	93.8	95.7	93.1
African ancestry							
multi-ethnic model	94.1	81.7	94.6	78.5	79.2	80.2	83.9
ethnic-specific model	92.4	76.8	88.5	77.1	80.0	79.4	74.2

don't tend to contribute to accuracy. For example, a threshold of 25 classifiers was used to filter out less important SNPs for European ancestry, and the total number of SNP predictors for *HLA–A*, *B*, ..., *DPB1* changes from 1042 to 779 without reducing accuracy (data not shown).

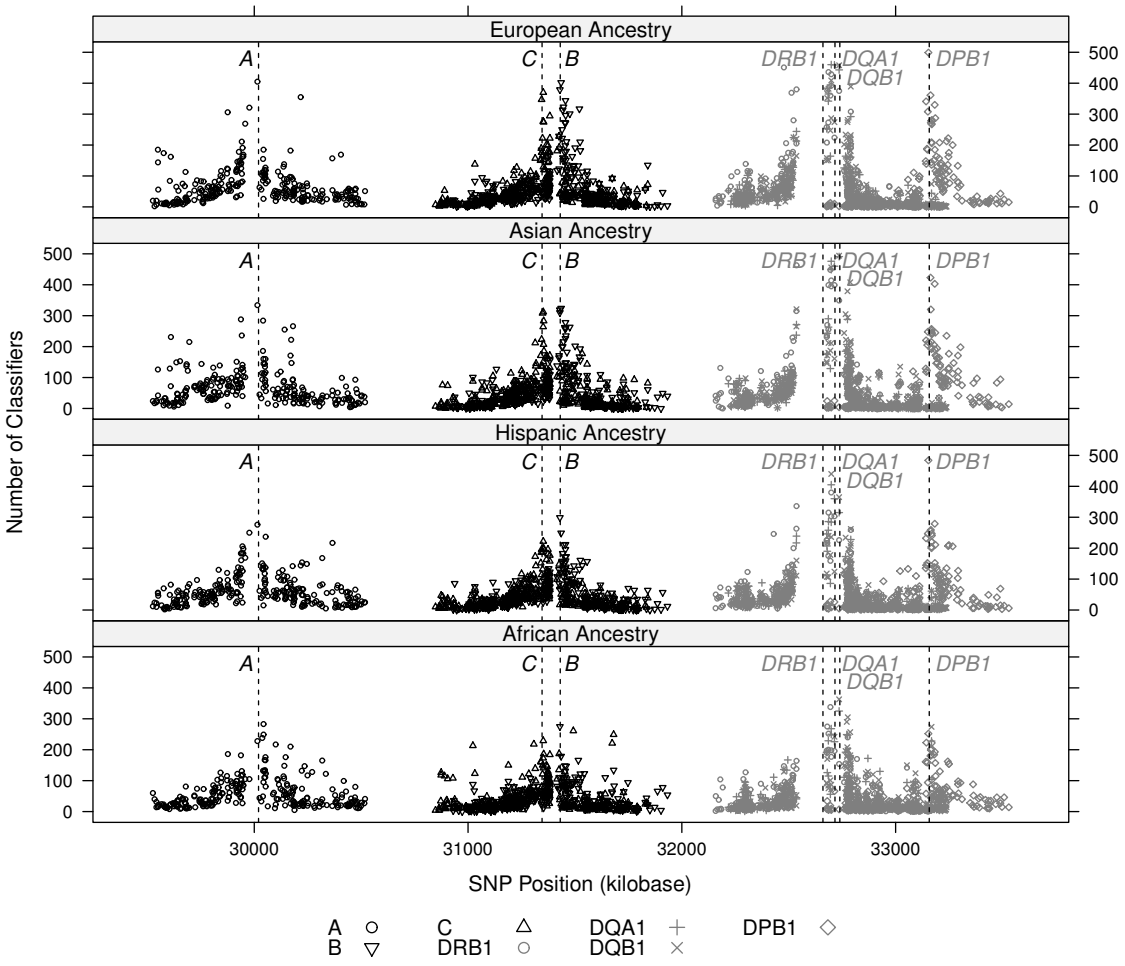


Figure 2.12: The number of classifiers used in the published pre-fit models for each SNP predictor. Each HIBAG model consists of 500 individual classifiers, and more important SNP markers tend to be used more frequently.

2.7 Discussion

We propose HIBAG, an ensemble classifier, for the imputation of HLA types from dense SNP data. The HIBAG classifier consists of many individual classifiers and makes a prediction by averaging HLA type posterior probabilities over the collection. Our comparisons indicate that HIBAG performs marginally better than HLA*IMP

developed by Leslie el at. (2008), and is comparable to BEAGLE. HIBAG prediction accuracies for individuals of European ancestry range from 94.8% to 99.2% when using a call threshold of 0.5 with a subset of SNPs common to several popular Illumina platforms.

Studies that identify significant associations within the MHC may be limited by the high cost of typing to investigate the contributions of underlying HLA alleles. Our SNP-based method provides an efficient way of imputing HLA types using genome-wide genotype data. A previous study has indicated that MHC-class-I-mediated events, principally involving *HLA–B*39*, contribute to the aetiology of type 1 diabetes [79]. HLA alleles are associated with some of the strongest adverse drug reactions, e.g. *B*57:01* with Abacavir which is used to treat HIV and AIDS [43], and *B*58:01* with Allopurinol used primarily to treat hyperuricemia [49]. Our results show that the predictions of *B*57:01* and *B*58:01* have 100% sensitivities and specificities with call rates over 95% for Europeans.

HIBAG produces the posterior probability of each HLA type. A direct application is to use the best-guess genotypes and call threshold in downstream association analysis, such as an additive logistic regression model [93]. As shown in Figure 2.6, individuals with higher posterior probabilities have a higher likelihood of being a correct call, and call threshold of 0.5 approximately corresponds to a prediction accuracy 80%. An alternative could be to model the uncertainty of prediction via posterior probabilities.

Another valuable application is to employ the HIBAG method in the design of HLA gene chips using SNP-specific hybridization probes for population-scale purposes. The extensive allelic diversity has made, and continues to make, high-resolution HLA DNA typing challenging [32]. With the advantage of detecting thousands of SNPs in

parallel, the oligonucleotide array technology has provided a cost-effective approach for high-throughput polymorphism analysis. Given a training data set, our method utilizes a statistical learn approach to build a connection between HLA types and SNPs, and it is not necessary that SNPs should be in an HLA gene.

Our method and parameter estimates are freely available in the HIBAG R package. A typical parameter file for imputing HLA types contains only haplotype frequencies at different SNP subsets rather than individual training genotypes. Further, unlike the web-implemneted HLA*IMP, HIBAG does not require the uploading of genotype information to a website, which could raise concerns over data privacy, or having access to large training HLA datasets. To facilitate future use of this method, we have prepared pre-fit classifiers based on our STUDY Data, which can be used to impute HLA types in new SNP data. The SNP markers selected in the pre-fit classifiers are common to the Illumina 1M Duo, OmniQuad, OmniExpress, 660K and 550K platforms within a 500kb flanking region of each HLA gene. A complementary validation study on our published pre-built classifiers of European ancestry is shown in Supplementary Table 2.10, using four-digit HLA data from the British 1958 birth cohort as test samples. A comparison of accuracies in Table 2.7, 2.9 and 2.10 indicates the study data and method are robust and consistent.

In our published pre-fit classifiers, we selected 1042 SNPs in total. However, it is possible to find a smaller set of SNPs without sacrificing accuracy using our method. HLA*IMP [28] has developed a selection approach to identify a small set of most informative SNPs for predicting HLA types. Because of some high LD across the xMHC, it is possible to identify a second or third prediction set of almost equal quality with little or no SNP overlap [59]. With respect to HIBAG, variable selection is implicitly incorporated during the construction of each individual classifier, and

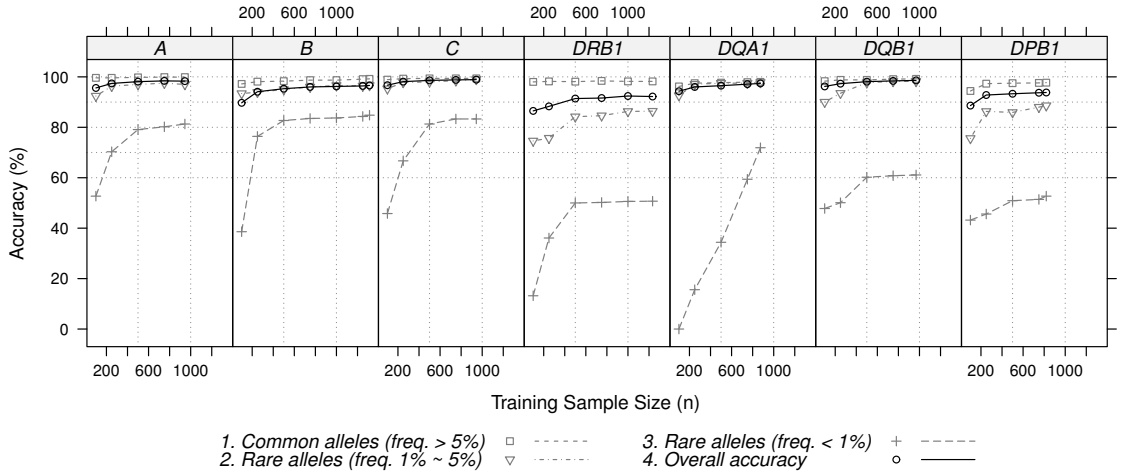


Figure 2.13: The relationship between training sample size and accuracy. Study data of European ancestry were divided into training and validation sets with approximately equal sizes, and random subsets of training samples ($n = 100, 250, 500, 750, 1000$ and max) were used to build a HIBAG model which was applied to the same validation samples. No call threshold was applied.

more important SNP markers tend to be used more frequently in the ensemble.

When STUDY Data of European ancestry was investigated, the overall accuracies increase with the training sample size, but are only slightly improved after 500 training samples, as shown in Figure 2.13. The rare alleles of $< 1\%$ frequency have significantly lower prediction accuracies than the common alleles. The size of sample sets required to accurately type rare alleles using an imputation methodology, is impractical. While we observed 144 unique *HLA–B* alleles in our total study population ($n = 5515$), typing of $> 28,000$ individuals for *HLA–B* by the Nation Marrow Donor Program [68] only identified 184 unique *HLA–B* alleles, still representing less than 10% of the 1898 four digit *HLA–B* alleles currently identified by IMGT.

The accuracies of common alleles for *HLA–A*, *B*, *C* and *DQB1* are higher than 99%, whereas that of *DPB1* is the lowest ($\sim 97\%$). Possible reasons for imperfect pre-

dictions on the alleles of $> 1\%$ frequency are data quality of genotypes, the ambiguity of HLA alleles due to typing approach, missing SNPs, and loss of distinguishable SNP patterns. Leslie et al. (2008) did observe chromosomes that have nearly identical SNP patterns, yet carry different HLA alleles [59]. Denser SNP markers, especially those SNPs in an HLA gene, may increase overall accuracies.

HLA*IMP relies on high-quality haplotypes in the training data [59], which contain the HLA locus of interest and SNP predictors. However, most experimental techniques for determining SNPs do not provide haplotype information, and the quality of computational phasing of unrelated individuals may not be satisfactory. Our result shown in Figure 2.10 indicates that it is possible to improve prediction accuracy by taking uncertainty of phasing into account. On the other hand, BEAGLE assumes variable-length Markov chains besides HWE to represent linkage disequilibrium [14], which is a bias-variance tradeoff in a possibly very high-dimensional problem [18]. Since linkage disequilibrium in the xMHC typically follows a complex pattern, HIBAG does not make any assumption except HWE and is possibly more suitable to the complex MHC region than methods with additional assumptions.

It is important to realize the potential limitations and our findings should be interpreted with caution. The numbers of HLA alleles documented in the IMGT-HLA database [94] are much larger than the numbers investigated in our study. For example, the numbers of four-digit HLA alleles from IMGT are 1365, 1898 and 1006 at the *HLA–A*, *B* and *C* loci and new alleles are routinely being discovered, but we only have 85, 144 and 49 alleles in our training samples. The prediction accuracies reported here are computed from restricted validation samples whose HLA alleles are present in the training set. Quite large training sets are required to successfully predict most of HLA alleles in the IMGT-HLA dataset, since ten copies of an allele

in the training database are generally required to provide high sensitivity [59].

In summary, we propose a new method for HLA type imputation with performance similar to existing methods including HLA*IMP and BEAGLE with several differentiating factors. The HIBAG and BEAGLE utilize all the available SNPs in the region which result in increased accuracy for these methods versus HLA*IMP. The freely available HIBAG method and accompanying parameter estimates (published in this paper) enable the method to be applied without the need to upload data to an external website (i.e., HLA*IMP) or to have access to a training dataset (BEAGLE).

Chapter 3

Relatedness Analysis in Genome-Wide Association Studies

3.1 Abstract

Principal component analysis (PCA) is widely used in genome-wide association studies (GWAS) to detect and adjust for population structure, and the principal component axes often represent perpendicular gradients in geographic space. However, PCA is a model-free approach and seems like a "black box". Here, I provided an interpretation of PCA based on relatedness measures, which are described by the probability that sets of genes are identical-by-descent (IBD). An approximately linear transformation between the projection of individuals onto the principal components and allele admixture fractions assuming two or more ancestral populations was revealed.

Identification of population structure is of great interest for genetic association studies. Here, I proposed an individual dissimilarity measure to be used in hierarchical cluster analysis to identify clusters of individuals. Compared to other dissimilarity measures, its expected value is directly related to the coancestry coefficient (or kinship coefficient) without being confounded by allele frequency, and it could be used to estimate the kinship coefficient if a reference population can be identified. Also, the individual dissimilarity measure is a moment estimator and suited for large-scale GWAS data.

Both of PCA and hierarchical cluster analysis were applied to HapMap Phase II and III data. The population admixture proportions inferred by PCA were well consistent with what HapMap Phase III has reported. Hierarchical clustering analysis splitted study individuals based on their coancestry coefficients, and successfully separated Chinese and Japanese samples in HapMap Phase II data. Finally, a combination of PCA and hierarchical cluster analysis should help us better understand population structure for isolated and admixed populations.

3.2 Introduction

Genome-wide association studies (GWAS) are widely used to investigate the genetic basis of many complex diseases and traits, but the large volumes of data generated in GWAS from thousands of study samples pose significant analytical challenges. One important challenge is the inflated false-positive associations that arise in GWAS results when population structure and cryptic relatedness exist [19, 21]. These analytical challenges could be addressed by using principal component analysis (PCA) to detect and correct for population structure [83, 86] and identity-by-descent (IBD) methods to identify the degree of relatedness between each pair of study samples [21].

3.2.1 Principal Component Analysis

Principal component analysis was first introduced to the study of genetic data almost thirty years ago by Cavalli-Sforza [76], and has become a standard tool in genetics. Population differentiation was inferred from multivariate statistical methods like PCA using allele frequencies to reflect the geographical distribution of populations [76, 20]. Unlike previous applications, Patterson *et al.* (2006) applied PCA to SNP genotypic data of individuals rather than populations. Their method was implemented

in a software package "EIGENSTRAT", which was primarily designed to correct for population stratification in genetic association studies. However, PCA is not based on any population genetics model, and seems like a "black box" method [83], although the principal component axes often represent perpendicular gradients in geographic space [20, 81, 82, 86]. Novembre & Stephens (2008) suggests that the approach of adjusting for the largest principal components in an association study is conceptually similar to modeling smooth geographical trends in phenotype means for spatially continuous populations.

From the perspective of coalescent theory, McVean (2009) provided a genealogical interpretation of PCA. He showed that the projection of samples onto the principal components can be obtained directly from the pairwise coalescence times between samples [72]. He also indicate that the projection of admixed individuals onto existing axes directly identifies admixture fractions. A link between PCA and Wright's F_{st} was demonstrated.

Here, I provide an alternative interpretation of PCA based on relatedness measures, which are described by the probability that sets of genes have descended from a single ancestral gene, i.e., the probability that they are identical-by-descent (IBD). In population genetics, Weir & Hill (2002) [113] extended the work of Weir & Cockerham (1984) [112] by allowing different levels of coancestry for different populations, and by allowing non-zero coancestries between pairs of populations. A possible further extension is to allow different coancestries between pairs of individuals and different inbreeding coefficients for individuals, or to use the more comprehensive set of nine Jacquard IBD measures [51]. These individual-perspective measures of population structure can be used to explain PCA. A link between coalescent time and inbreeding coefficient of a population (or F_{st}) has been revealed [72, 100], hence my interpre-

tation of PCA should be similar to McVean's. The primary difference between my approach and that of McVean is that I formally prove a link between the projection of samples onto the principal components and allele admixture fractions (AF) assuming two or more ancestral populations, whereas McVean (2009) described it briefly. My derivation also reveals a theoretical bias when inferring AF from the projection of samples, whereas McVean (2009) did not point out such bias.

Allele admixture fractions of an individual refer to the fractions of the genome derived from ancestral populations [35, 90, 103]. The early approach for estimating AF can track back to Hanis *et al.* [40], and the ancestral allele frequencies should be known to allow estimating allele admixture in this method. However, the ancestral allele frequencies are usually estimated from pseudo-ancestral samples in practice. The studies cited took into account the uncertainty of estimating ancestral information by pseudo-ancestor samples to develop their approaches.

A Bayesian approach, STRUCTURE, was developed to infer population substructure using unlinked genotypes [90]. Later, it was extended to model linked markers [35]. STRUCTURE is extremely computationally intensive and not likely to be suitable for large-scale studies, like GWAS, involved with thousands of individuals and hundreds of thousands of SNPs. Pruning SNPs has to be done before applying STRUCTURE, but this can introduce selection bias with respect to different SNP sets. A maximum-likelihood estimation method (frappe) [103] was also proposed to estimate AF and requires much less computation than STRUCTURE; however, frappe assumes the markers are unlinked although it also works when linkage disequilibrium is weak. The program "ADMIXTURE" was further developed to analyze thousands of markers, and its modeling was similar to STRUCTRUE making an assumption of linkage equilibrium among the markers [4]. Recently, detection of chromosomal seg-

ments in admixed populations has been of great interest: HAPMIX was proposed to infer local ancestry from dense SNP markers based on approximate coalescent models modeling linkage disequilibrium [87], and it also allowed estimation of population admixture proportions [50]. In addition, AF could be estimated from the largest principal components. In this study, I describe an approach to estimate AF from PCA, and empirically compare the results with those from other software.

3.2.2 Relatedness Analysis

To adjust for population structure in genetic association studies, an alternative is to employ estimated pairwise relatedness [21]. Jacquard (1972) described a set of nine identity-by-descent (IBD) modes that give a full description of the possible IBD relationships between the set of four alleles of two individuals, assuming symmetry between maternal and paternal gametes, as shown in Figure 3.1. If it is assumed that two related individuals are not inbred, then only three IBD modes are possible: S_7, S_8 and S_9. Otherwise, the other IBD modes can occur when one or both of the two individuals are inbred. As an example, if two non-inbred individuals are full siblings (that is, they share both a mother and a father and the mother and father are unrelated), then $\Delta_7 = 1/4$, $\Delta_8 = 1/2$, and $\Delta_9 = 1/4$. All other IBD modes are impossible for noninbred full siblings, i.e., $\Delta_1 = \cdots = \Delta_6 = 0$.

Estimation of the probability that two alleles are identical-by-descent requires a reference population. As shown in Figure 3.2, referring to the population at t_3, none of the three alleles labeled 1, 2 and 3 are IBD. However, when the population at t_1 is treated as a reference, all of them are IBD since they are derived from the same allele at t_1. When the population at t_2 is considered, allele 1 and 2 are inherited from the same ancestral allele, but allele 3 is not.

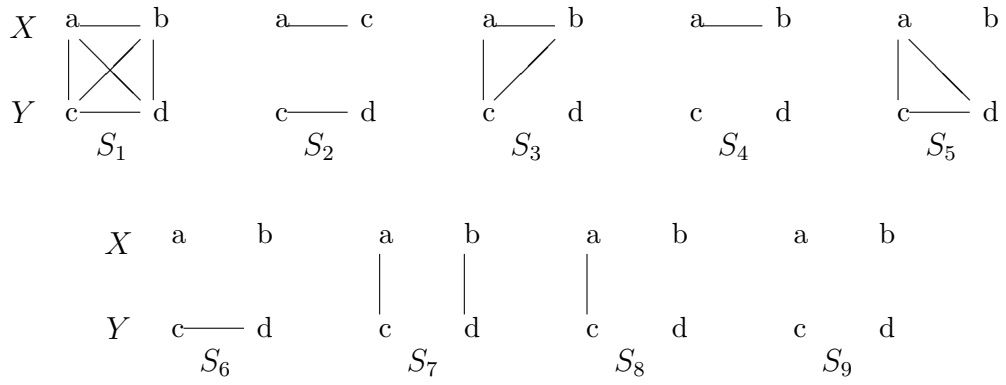


Figure 3.1: Graphic representation of the nine IBD patterns of Jacquard. Solid lines join the IBD pairs of alleles. X has alleles a, b and Y has alleles c, d. The probability of state S_i is Δ_i.

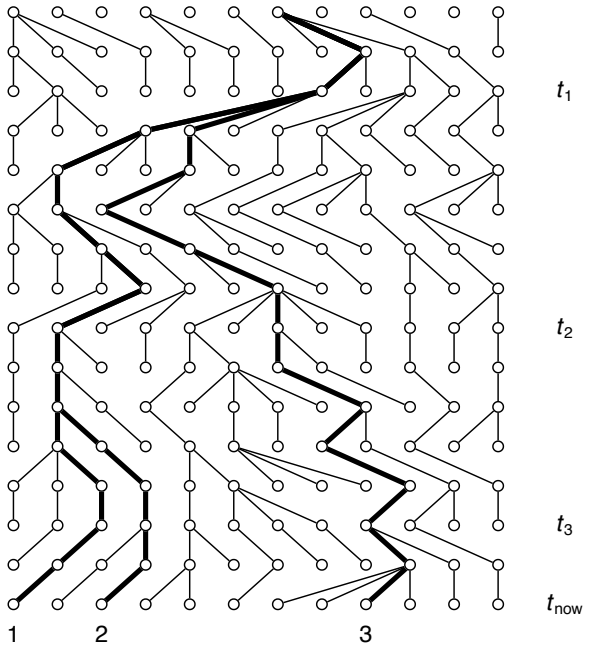


Figure 3.2: The genealogy of three sampled alleles in a haploid reproductive model. The ancestry of the alleles are marked by bold lines sixteen generations back in time.

A potential problem arises in relatedness (IBD) analysis when the allele frequencies that are needed for estimating the relationship vary among subpopulations due to population structure. This leads to a problem of miss-specified allele frequencies. The effect of using allele frequencies from the overall population to estimate the relatedness of two individuals in a subpopulation, is conceptually similar to shifting the reference from which we measure relatedness back in time [5], as an example "$t_3 \rightarrow t_2 \rightarrow t_1$" in Figure 3.2. The two individuals in the subpopulation would be more related relative to the whole population.

Two methods have been widely used to estimate the three IBD coefficients for pairs of individuals in a non-inbred or homogeneous population: maximum likelihood estimation (MLE) [21, 105] and Method of Moments (MoM), as implemented in PLINK [91]. However, allele frequencies need to be specified for both methods, whereas they are usually estimated from a study sample. Anderson and Weir (2007) proposed a maximum-likelihood method for estimating pairwise relatedness in structured populations [5], in which an inbreeding coefficient F_{st} is included in the likelihood function for a pair of genotypes. Wang (2011) proposed a complementary method of moment for estimating kinship coefficients in structured populations [111]. However, F_{st} needs to be specified in that method also, and thus is usually unknown when population structure is not identified.

Clustering methods based on PCA and identity-by-state (IBS) have been proposed to group study individuals [37, 58, 83]. PCA clustering methods tend to be less efficient when not all of principal components are used, since PCA is primarily designed for dimension reduction. The expected value of IBS-based dissimilarity is confounded by allele frequency and it is difficult to interpret the distance or height in a dendrogram (see Section 3.3.6). Based on the individual perspective for measures

of population structure, I propose an individual dissimilarity measure to be used in hierarchical cluster analysis to identify clusters of individuals. Compared to PCA- and IBS-based measures the expected value of the dissimilarity estimate is directly related to the coancestry coefficient with an appropriate interpretation from a genetic perspective. A clustering algorithm based on a dendrogram and a Z score computed from permutation of individuals is proposed to group study individuals. A combination of PCA and hierarchical cluster analysis will help us understand and identify population structure in real data.

The outline of this chapter is as follows: Section 3.3.1 and 3.3.2 derive the expected values of Patterson's PCA correlation matrix and individual dissimilarity from both population and individual perspectives. Section 3.3.3 analyzes the eigen-decomposition of PCA and points out a link between largest principal components and allele admixture fractions. A hierarchical clustering with individual dissimilarity is described in Section 3.3.6. The application of developed R package "SNPRelate" [120] to HapMap Phase II & III data was conducted in the result section (Section 3.4). Finally, a summary is provided in Section 3.5.

3.3 Methods

To distinguish the populations sampled, an index i is added to the indicator variables for the i^{th} sample. Let an index j denote the j^{th} study individual samples from a specific population, and l denote the l^{th} locus. Let N_P be the total number of populations, n_i be the sample size for the i^{th} population, and $n = n_1 + ... + n_{N_P}$. Let L denote the total number of loci scored. The mathematical symbols used in this chapter are described in Table 3.1.

Table 3.1: Summary of mathematical symbols used in Chapter 3.

Symbol	Description
an index i	the i^{th} population sampled
an index j	the j^{th} study individual in a specific population
an index k	$k \in \{1, 2\}$, indicating which chromosome, but which chromosome is defined as the first one is arbitrary
an index l	the l^{th} SNP locus
x	a SNP allele indicator ($x = 1$ if it is A allele, otherwise $x = 0$)
g	a SNP genotype indicator ($g \in \{0, 1, 2\}$), e.g., $g_{ijl} = x_{ij1l} + x_{ij2l}$
p_l	the allele frequency for A allele at the l locus
L	the total number of SNPs
N_P	the total number of populations
n_i	the sample size of the i^{th} population, $1 \leq i \leq N_P$
n	the total number of individuals, $n = \sum_{i=1}^{N_P} n_i$
θ_W	$(\sum_{i=1}^{N_P} n_i\theta_i)/(\sum_{i=1}^{N_P} n_i)$, the weighted average inbreeding coefficient (θ_i) within populations
θ_A	$(\sum_{i\neq i'}^{N_P} n_i n_{i'}\theta_{ii'})/(\sum_{i\neq i'}^{N_P} n_i n_{i'})$, the weighted average coancestry coefficient ($\theta_{ii'}$) among (A) populations
θ_I	$= \frac{1}{n}\sum_{j=1}^{n}\theta_j$, the average inbreeding coefficient (θ_j) across all loci among all study individuals
θ_T	$= \frac{1}{n(n-1)}\sum_{j,j'=1,j\neq j'}^{n}\theta_{jj'}$, the average kinship coefficient ($\theta_{jj'}$) across all loci among all study individuals
ψ_j	$= \frac{1}{n}\sum_{j'=1}^{n}\theta_{jj'}$ (let $\theta_{jj} = \theta_j$)
N	the number of ancestral populations
$\mathbf{r}_j$	$= (r_{j,1}, ..., r_{j,N})$, allele admixture fractions of individual j

In principal component analysis, Patterson *et al.* (2006) estimates genetic correlation matrix by $\mathbb{M}^{\text{P}} = [m^{\text{P}}_{j,j'}]$, where

$$m^{\text{P}}_{j,j'} = \frac{1}{L}\sum_{l=1}^{L}\frac{(g_{jl} - 2\bar{p}_l)(g_{j'l} - 2\bar{p}_l)}{\bar{p}_l(1-\bar{p}_l)} \tag{3.1}$$

and $\bar{p}_l = \frac{1}{2n}\sum_{j=1}^{n} g_{jl}$ is the sample allele frequency at the l locus. The numerator is the product of mean-adjusted number of reference allele (0, 1 or 2 − mean over individuals). The denominator is motivated by the fact that the frequency change of

a SNP due to genetic drift occurs at a rate proportional to $\sqrt{p_l(1-p_l)}$ per generation [83]. It also normalizes each SNP to have the same variance if that SNP is in Hardy-Weinberg equilibrium [83].

In this chapter, the proposed measure of individual dissimilarity is defined as

$$d_{j,j'} = \begin{cases} \dfrac{\sum_{l=1}^{L} g_{jl}(2-g_{jl})}{2\sum_{l=1}^{L}\bar{p}_l(1-\bar{p}_l)} & \text{, if } j = j' \\ \\ \dfrac{\sum_{l=1}^{L} g_{jl}(2-g_{j'l}) + g_{j'l}(2-g_{jl})}{8\sum_{l=1}^{L}\bar{p}_l(1-\bar{p}_l)} & \text{, if } j \neq j' \end{cases} \tag{3.2}$$

The expected value of individual dissimilarity is directly related to coancestry coefficient (or kinship coefficient) without being confounded by allele frequency. It is also a moment estimator and suited for large-scale GWAS data. The derivation of the formulas is provided in Section 3.3.1 and 3.3.2.

In sections 3.3.1 and 3.3.2, I discuss the expected values of $m^{\mathrm{P}}_{j,j'}$ and $d_{j,j'}$ from two interrelated perspectives of relatedness: population coancestry and relatedness with nine Jacquard's coefficients. These two perspectives lead to the same expected values of $m^{\mathrm{P}}_{j,j'}$ and $d_{j,j'}$. The next section 3.3.3 finds a connection between eigen-decomposition of PCA and allele admixture fractions. A method of hierarchical clustering based on the proposed individual dissimilarity is described in Section 3.3.6.

3.3.1 Population Coancestry Framework of Weir & Hill (2002)

Let

$$x_{ijkl} = \begin{cases} 1 & \text{the } k^{\text{th}} \text{ allele in an individual } j \text{ from the } i^{\text{th}} \text{ population} \\ & \text{is type A at locus } l \\ 0 & \text{otherwise} \end{cases}$$

$k \in \{1, 2\}$ and g_{ijl} denotes the SNP genotype at locus l for the individual j in the i^{th} population, $g_{ijl} = x_{ij1l} + x_{ij2l} \in \{0, 1, 2\}$.

Under the framework of Weir & Hill (2002), assuming no local inbreeding, expectations for first and second moments of the xs are

$$\begin{aligned}
\mathcal{E}[x_{ijkl}] &= p_l \\
\mathcal{E}[x_{ijkl}^2] &= p_l \\
\mathcal{E}[x_{ij1l}\ x_{ij2l}] &= p_l^2 + p_l(1-p_l)\theta_i \quad \text{, the same individual} \\
\mathcal{E}[x_{ijkl}\ x_{ij'k'l}] &= p_l^2 + p_l(1-p_l)\theta_i \quad \text{, } j \neq j' \text{, the same population} \\
\mathcal{E}[x_{ijkl}\ x_{i'j'k'l}] &= p_l^2 + p_l(1-p_l)\theta_{ii'} \quad \text{, } i \neq i' \text{, different populations}
\end{aligned}$$

given by inbreeding coefficients θ_i and coancestry coefficients $\theta_{ii'}$, where p_l is the frequency of allele A in the reference population, and all study individuals are traced back to a single reference population. This reference population could be common ancestors at a point in time. The equal values for $\mathcal{E}[x_{ij1l}\ x_{ij2l}]$ and $\mathcal{E}[x_{ijkl}\ x_{ij'k'l}]$ require an assumption of random mating.

The inbreeding coefficient θ_i refers to the probability of identity by descent (IBD) for a random pair of alleles in population i, and the pair of alleles can come from the same individual. The coancestry coefficient $\theta_{ii'}$ refers to the probability of IBD for a random pair of alleles when one allele is from population i and the other is from population i'. Note that we implicitly assume that θ_i and $\theta_{ii'}$ are the same at each locus, and in practice θ_i and $\theta_{ii'}$ are actually the average inbreeding and coancestry coefficients over all L loci.

Weir & Hill (2002) wrote θ_W and θ_A for the weighted average inbreeding coefficients within (W) populations and weighted average coancestry coefficient among (A)

populations, where

$$\begin{aligned} \theta_W &= (\textstyle\sum_{i=1}^{N_P} n_i\theta_i)/(\sum_{i=1}^{N_P} n_i) \\ \theta_A &= (\textstyle\sum_{i\neq i'}^{N_P} n_i n_{i'}\theta_{ii'})/(\sum_{i\neq i'}^{N_P} n_i n_{i'}) \end{aligned} \tag{3.3}$$

Now consider an individual perspective measures of population structure, i.e., a special case of Weir & Hill's population coancestry framework: each population i has only one sampled individual ($n_i = 1$) so $j = 1$ for each population. The sample size n is also the number of populations N_P. Therefore,

$$\begin{aligned} \bar{p}_l &= \tfrac{1}{N_P}\textstyle\sum_{i=1}^{N_P}\bar{p}_{il} = \tfrac{1}{n}\sum_{i=1}^{n}[\tfrac{1}{2}\sum_{j=1}^{1}(x_{ij1l} + x_{ij2l})] \\ \mathcal{E}[\bar{p}_l] &= p_l \\ \mathrm{Var}[\bar{p}_{il}] &= \tfrac{1}{2}p_l(1-p_l)(1+\theta_i) \\ \mathrm{Cov}[\bar{p}_{il}, \bar{p}_{i'l}] &= p_l(1-p_l)\theta_{ii'} \\ \mathrm{Var}[\bar{p}_l] &= \tfrac{n-1}{n}p_l(1-p_l)\theta_A + \tfrac{1}{2n}p_l(1-p_l)(1+\theta_W) \\ \mathcal{E}[\bar{p}_l(1-\bar{p}_l)] &= \tfrac{n-1}{n}p_l(1-p_l)(1-\theta_A) + \tfrac{1}{2n}p_l(1-p_l)(1-\theta_W) \end{aligned} \tag{3.4}$$

Here these individual perspective measures do not account for pedigree data and the relatedness of individuals is established from evolutionary history.

Let us use θ_T and θ_I (defined in Table 3.1) to replace θ_A and θ_W in Equation 3.4, since θ_A and θ_W were defined in Weir & Hill (2002) for the purpose of a population, i.e.,

$$\mathcal{E}[\bar{p}_l(1-\bar{p}_l)] = \tfrac{n-1}{n}p_l(1-p_l)(1-\theta_T) + \tfrac{1}{2n}p_l(1-p_l)(1-\theta_I)$$

where θ_T and θ_I are defined from the perspective of an individual. For a large number of individuals,

$$\mathcal{E}[\bar{p}_l(1-\bar{p}_l)] = p_l(1-p_l)(1-\theta_T) \tag{3.5}$$

Patterson's PCA

The genetic correlation matrix is estimated by $\mathbb{M}^{\mathrm{P}}$ in Patterson *et al.* (2006). If each population has only one individual, then we simply drop the index j but keep i to distinguish populations (individuals). The expected values of the numerator in Equation 3.1 is:

$$\begin{aligned}\mathcal{E}[(g_{il}-2\bar{p}_l)^2] &= p_l(1-p_l)(2+2\theta_i+4\frac{n-1}{n}\theta_T-8\psi_i)\\ &\quad +\frac{2}{n}p_l(1-p_l)(\theta_I+2\theta_i-1) \qquad \text{, for } i=i'\end{aligned}$$

$$\begin{aligned}\mathcal{E}[(g_{il}-2\bar{p}_l)(g_{i'l}-2\bar{p}_l)] &= 4p_l(1-p_l)(\theta_{ii'}+\frac{n-1}{n}\theta_T-\psi_i-\psi_{i'})\\ &\quad +\frac{2}{n}p_l(1-p_l)(\theta_I+\theta_i+\theta_{i'}-1) \qquad \text{, for } i\neq i'\end{aligned}$$

where $\psi_i=\frac{1}{n}\sum_{i'=1}^{n}\theta_{ii'}$ (let $\theta_{ii}=\theta_i$).

When the number n of study individuals is large enough,

$$\mathcal{E}[\frac{1}{4}m^{\mathrm{P}}_{i,i'}]=\begin{cases}\dfrac{1+\theta_i}{2(1-\theta_T)}+\dfrac{\theta_T-2\psi_i}{1-\theta_T} & \text{, if } \ i=i'\\ \\ \dfrac{\theta_{ii'}}{1-\theta_T}+\dfrac{\theta_T-\psi_i-\psi_{i'}}{1-\theta_T} & \text{, if } \ i\neq i'\end{cases} \tag{3.6}$$

Individual Dissimilarity

The expected values of the numerators in Equation 3.2 are

$$\begin{aligned}\mathcal{E}[g_{il}(2-g_{il})] &= 2p_l(1-p_l)(1-\theta_i)\\ \mathcal{E}[g_{il}(2-g_{i'l})+g_{i'l}(2-g_{il})] &= 8p_l(1-p_l)(1-\theta_{ii'})\end{aligned}$$

Therefore, when n is large enough,

$$\mathcal{E}[d_{i,i'}] = \begin{cases} \frac{1-\theta_i}{1-\theta_A} & \text{, if } \quad i = i' \\ \frac{1-\theta_{ii'}}{1-\theta_A} & \text{, if } \quad i \neq i' \end{cases} \tag{3.7}$$

3.3.2 Relatedness with Jacquard's Coefficients

Jacquard (1972) described a set of nine identity-by-descent (IBD) modes that give a full description of the possible IBD relationships between the set of four alleles possessed by two (possibly inbred) individuals in the same population, as shown in Figure 3.1. For bi-allelic loci, the joint genotype probabilities for a pair of individuals are shown in Table 3.2.

Thereby, a set of expectations are

$$\begin{aligned} \mathcal{E}[g_{jl}] &= 2p_l \\ \mathcal{E}[g_{jl}^2] &= 4p_l^2 + 2p_l(1-p_l)(1+\theta_j) \\ \mathcal{E}[g_{jl}\ g_{j'l}] &= 4p_l^2 + 4p_l(1-p_l)\theta_{jj'} \qquad , j \neq j' \end{aligned}$$

where p_l is the frequency of allele A in the reference population, and all study individuals j are traced back to a single reference population. θ_j is an inbreeding coefficient for individual j, and $\theta_{jj'}$ is kinship coefficient between individuals j and j'. Unlike the previous section 3.3.1, there is no population specified to individuals, thereby we drop the index i from SNP genotype g.

Either directly from the previous section, by regarding the set of n individuals j in one sample $i = 1$ as being formally equivalent to a set of 1 individual from each of n populations, or by repeating the previous derivations, starting with the sample allele frequency at the l locus now written as $\bar{p}_l = \frac{1}{2n}\sum_{j=1}^{n} g_{jl}$, then the results given

Table 3.2: Joint genotypic probabilities for a pair of individuals X and Y using nine Jacquard's IBD coefficients $(\Delta_1, ..., \Delta_9)$ at a SNP locus [1].

Genotypes (X, Y)	(g_X, g_Y)	Probability
(AA, AA)	(2,2)	$\Delta_1 p + (\Delta_2 + \Delta_3 + \Delta_5 + \Delta_7)p^2 + (\Delta_4 + \Delta_6 + \Delta_8)p^3 + \Delta_9 p^4$
(BB, BB)	(0,0)	$\Delta_1(1-p) + (\Delta_2 + \Delta_3 + \Delta_5 + \Delta_7)(1-p)^2 + (\Delta_4 + \Delta_6 + \Delta_8)(1-p)^3 + \Delta_9(1-p)^4$
(AB, AB)	(1,1)	$2\Delta_7 p(1-p) + \Delta_8 p(1-p) + 4\Delta_9 p^2(1-p)^2$
(AA, AB)	(2,1)	$\Delta_3 p(1-p) + (2\Delta_4 + \Delta_8)p^2(1-p) + 2\Delta_9 p^3(1-p)$
(AB, AA)	(1,2)	$\Delta_5 p(1-p) + (2\Delta_6 + \Delta_8)p^2(1-p) + 2\Delta_9 p^3(1-p)$
(BB, AB)	(0,1)	$\Delta_3 p(1-p) + (2\Delta_4 + \Delta_8)p(1-p)^2 + 2\Delta_9 p(1-p)^3$
(AB, BB)	(1,0)	$\Delta_5 p(1-p) + (2\Delta_6 + \Delta_8)p(1-p)^2 + 2\Delta_9 p(1-p)^3$
(AA, BB)	(2,0)	$\Delta_2 p(1-p) + \Delta_4 p(1-p)^2 + \Delta_6 p^2(1-p) + \Delta_9 p^2(1-p)^2$
(BB, AA)	(0,2)	$\Delta_2 p(1-p) + \Delta_4 p^2(1-p) + \Delta_6 p(1-p)^2 + \Delta_9 p^2(1-p)^2$

[1]: p is the allele frequency of A at that SNP locus.
Inbreeding coefficient $\theta_X = \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4$, $\theta_Y = \Delta_1 + \Delta_2 + \Delta_5 + \Delta_6$, and kinship coefficient $\theta_{XY} = \Delta_1 + \frac{1}{2}(\Delta_3 + \Delta_5 + \Delta_7) + \frac{1}{4}\Delta_8$.

in Equations 3.6 and 3.7 hold. Now the conditions on populations i are replaced by conditions on individuals j.

3.3.3 Eigen-decomposition in PCA

If we are interested in $\frac{1+\theta_j}{2}$ and $\theta_{jj'}$, the factors $(1-\theta_T)$ and $\frac{\theta_T - \psi_j - \psi_{j'}}{1-\theta_T}$ will confound the estimates when $\frac{1}{4}m^{\mathrm{P}}_{j,j'}$ is used. This may explain why a large proportion of $m^{\mathrm{P}}_{j,j'}$ are negative, while the true θ_j and $\theta_{jj'}$ are always between zero and one.

The Population Perspective

PCA conducts eigen-decomposition on the matrix $\mathbb{M}^{\mathrm{P}}$. To illustrate what eigen-decomposition does, I introduce a genetic model consisting of populations at three points in time as shown in Figure 3.3. The alleles of all study individuals at t_{now} can be tracked to a single reference population at t_0 through at least one of distinct

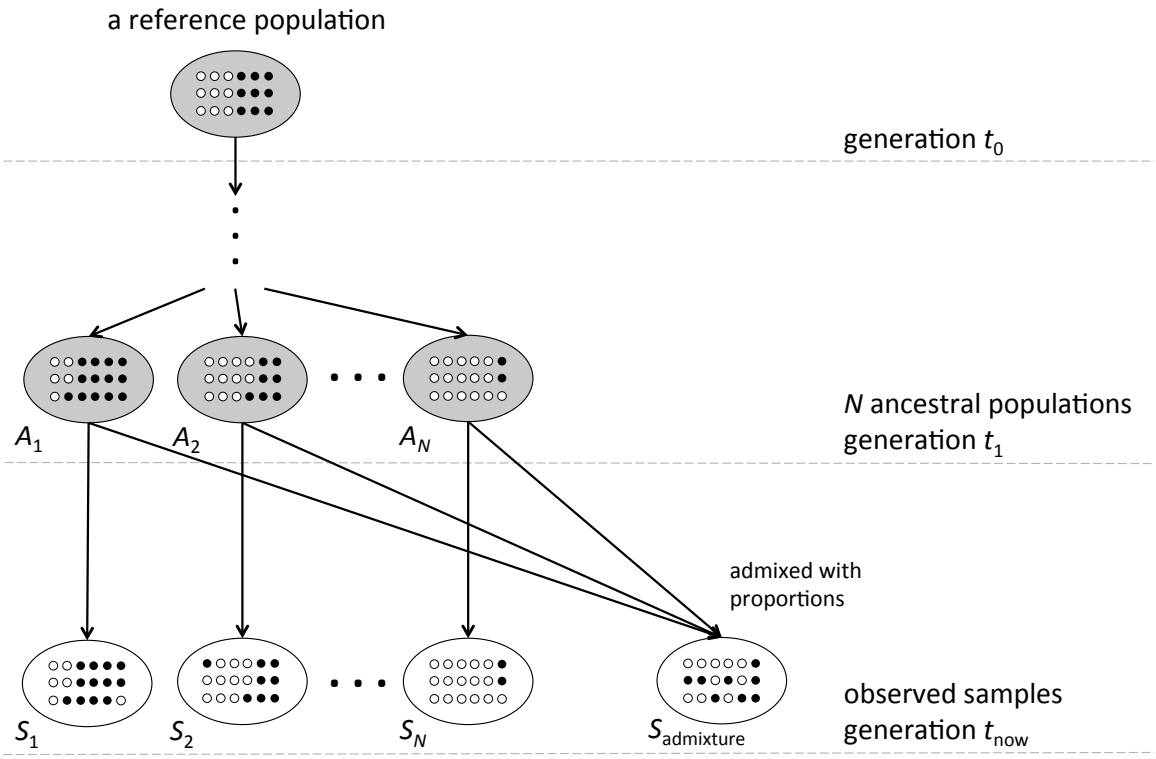


Figure 3.3: A genetic model at a single locus for observed samples. The alleles of all study individuals at t_{now} can be tracked to a single reference population at t_0, and there are N distinct ancestral populations at t_1. The relationships among ancestral populations are described by a coancestry matrix Θ_A.

ancestral populations at t_1. The study samples $S_1, ..., S_N$ are directly inherited from the ancestral populations $A_1, .., A_N$ without admixture, and the sample $S_{\text{admixture}}$ is admixed from N ancestral populations.

What we can observe are the genomes of study individuals at t_{now}. It could be appropriate to assume there are N ancestral populations at t_1 which is between t_0 and t_{now}, and the samples $S_1, ..., S_N$ are good candidates (or pseudo-ancestors) to represent the ancestral populations. For example, in the initial phase of the HapMap Project, genetic data were gathered from four populations (CEU, YRI, CHB and JPT) with European, African and Asian ancestry respectively. Here, $N = 3$, S_1 represents CEU

individuals, S_2 for YRI and S_3 for CHB+JPT.

According to the work of Weir & Hill (2002), a coancestry matrix Θ_A is used to describe the relationships among N ancestral populations at t_1 based on population perspective measures, where

$$\Theta_A = \begin{bmatrix} \theta_1^* & \theta_{12}^* & \cdots & \theta_{1N}^* \\ \theta_{12}^* & \theta_2^* & \cdots & \theta_{2N}^* \\ \cdots & \cdots & \ddots & \cdots \\ \theta_{1N}^* & \theta_{2N}^* & \cdots & \theta_N^* \end{bmatrix} \tag{3.8}$$

That is, θ_i^* is the average inbreeding coefficient of subjects in the i^{th} ancestral population, and $\theta_{ii'}^*$ is the average coancestry coefficient between the i^{th} and i'^{th} ancestral populations. Since we track all individuals back to the reference population at t_0, the sample allele frequencies at t_1 are treated as random variables over a probability space, which starts from the reference population at t_0 and arrives at the coancestry state Θ_A at t_1.

Allele Admixture Fraction

In practice individuals may have recent ancestors in more than one population. To model this, admixture models have been introduced, in which each individual is assumed to have inherited some proportion of its ancestry from each population [90, 35, 103]. Allele admixture fractions (AFs) represent proportions of an individual genome derived from ancestral populations.

For an individual j, let the AF be $\mathbf{r}_j = (r_{j,1}, ..., r_{j,N})^T$, where $\sum_{h=1}^{N} r_{j,h} = 1$ and $0 \leq r_{j,h} \leq 1$. Let $Z_{jklh} = 1$ when the k^{th} allele of individual j at SNP l is inherited from the h^{th} ancestral population, and $Z_{jklh} = 0$ otherwise. $\mathbf{Z}_{jkl} = \{Z_{jkl1}, ..., Z_{jklN}\}^T$

is a random variable with probabilities $\mathbf{r}_j$. That is

$$\mathcal{E}[Z_{jklh}] = r_{j,h}$$

or,

$$r_{j,h} = \mathcal{E}\left[\frac{1}{L}\sum_{l=1}^{L} Z_{jklh}\right]$$

representing the proportion of genome. We assume that the two alleles in individual j at SNP l are Bernoulli random variables (either A or B allele) that are independent and identically distributed (i.i.d), conditional on $\mathbf{r}_j$. This assumption has been made by other commonly used models of population structure, such as the Balding-Nichols model with admixture [6, 35, 103, 106].

Then the expected value of the inbreeding coefficient at SNP l for individual j is $\mathcal{E}[\mathbf{Z}_{j1l}^T \Theta_A \mathbf{Z}_{j2l}] = \mathbf{r}_j^T \Theta_A \mathbf{r}_j$, the same for each SNP. The average inbreeding coefficient over L loci is also $\theta_j = \mathbf{r}_j^T \Theta_A \mathbf{r}_j$, assuming the coancestry matrix of ancestral populations is identical at each locus.

For a pair of individuals j and j', we assume that any pair of alleles, one from j and the other from j' are independent Bernoulli random variables. Then the expected value of the kinship coefficient at SNP l is $\mathcal{E}[\mathbf{Z}_{jkl}^T \Theta_A \mathbf{Z}_{j'k'l}] = \mathbf{r}_j^T \Theta_A \mathbf{r}_{j'}$, and the average kinship coefficient over L loci is also $\theta_{jj'} = \mathbf{r}_j^T \Theta_A \mathbf{r}_{j'}$. This assumption is appropriate to model relatedness in structured population with admixture, and it has been used by other work like Thornton *et al.* (2012) [106], and treat $\mathbf{r}_j^T \Theta_A \mathbf{r}_{j'}$ as background relatedness due to evolutionary history. However, the assumption's validity is violated if individuals j and j' are in a family, e.g., parent and offspring.

Decomposition with Allele Admixture Fraction

For a study sample, there are n individuals in total, and no families. Each individual j has AF $\mathbf{r}_j$ with respect to N ancestral populations. Let $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n]^T$. Then the coancestry matrix of study individuals can be expressed as

$$\Theta_S \;=\; \mathbf{R}\;\Theta_A\;\mathbf{R}^T \tag{3.9}$$

We rewrite Equation 3.6 in matrix notation for large n,

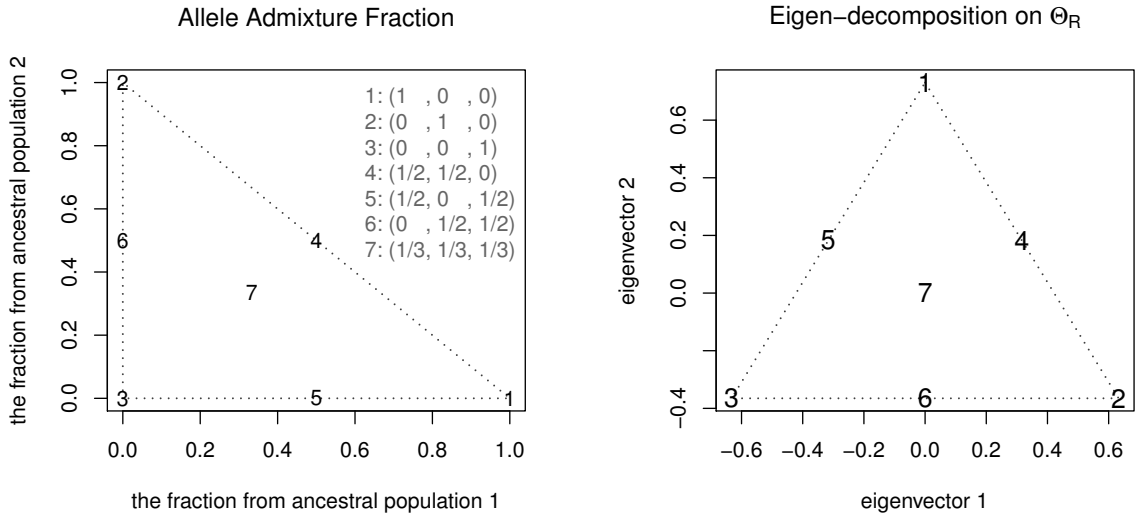
$$\mathcal{E}[\mathbb{M}^{\mathrm{P}}] = \frac{4}{1-\theta_T}\underbrace{(\mathbf{R}-\tfrac{1}{n}\mathbf{1}_{nn}\mathbf{R})\Theta_A(\mathbf{R}-\tfrac{1}{n}\mathbf{1}_{nn}\mathbf{R})^T}_{\stackrel{\text{def}}{=}\Theta_{\mathbf{R}}} + \underbrace{\mathrm{diag}(\tfrac{2(1-\theta_1)}{1-\theta_T},...,\tfrac{2(1-\theta_n)}{1-\theta_T})}_{\text{bias}} \tag{3.10}$$

where $\mathbf{1}_{nn}$ is a matrix of dimension $n \times n$ with entries equal to one. Since,

$$\begin{aligned}(\tfrac{1}{n}\mathbf{1}_{nn}\mathbf{R})\Theta_A(\tfrac{1}{n}\mathbf{1}_{nn}\mathbf{R})^T &= \tfrac{1}{n^2}\mathbf{1}_{nn}\Theta_S\mathbf{1}_{nn} = \tfrac{1}{n^2}(\textstyle\sum_{j\neq j'}\theta_{jj'} + \sum_j \theta_j)\mathbf{1}_{nn}\\ &= \left(\theta_T\tfrac{n(n-1)}{n^2} + \tfrac{1}{n}\theta_I\right)\mathbf{1}_{nn}\\ &\approx \theta_T\mathbf{1}_{nn}\end{aligned}$$

$$(\tfrac{1}{n}\mathbf{1}_{nn}\mathbf{R})\Theta_A\mathbf{R}^T = \tfrac{1}{n}\mathbf{1}_{nn}\Theta_S = \begin{bmatrix}\psi_1 & \psi_2 & \dots & \psi_n\\ \vdots & \vdots & \ddots & \vdots\\ \psi_1 & \psi_2 & \dots & \psi_n\end{bmatrix}.$$

Note that $\mathrm{rank}(\mathbf{R}-\frac{1}{n}\mathbf{1}_{nn}\mathbf{R}) \leq N-1$, and the largest $N-1$ principal components (PCs) of $\Theta_{\mathbf{R}}$ form a new coordinate with $N-1$ dimensions while AFs form an old N-dimensional coordinate. The mapping from the old coordinate to the new one is a linear transformation, and the proof is given in Section 3.3.4. Meanwhile, $\mathrm{diag}(\frac{2(1-\theta_1)}{1-\theta_T},...,\frac{2(1-\theta_n)}{1-\theta_T})$ is considered as a bias in the PCA with respect to AFs.



(a) (b)

Figure 3.4: The relationship between allele admixture fractions and eigen-decomposition: a) seven admixture fractions from three ancestral populations are plotted in the figure; b) the first and second eigenvectors of matrix $\Theta_{\mathbf{R}} = (\mathbf{R} - \frac{1}{n}\mathbf{1}_{nn}\mathbf{R})\Theta_A(\mathbf{R} - \frac{1}{n}\mathbf{1}_{nn}\mathbf{R})^T$, where Θ_A is assumed to $\text{diag}(0.05, 0.05, 0.05)$. The mapping from the coordinate in (a) to that of (b) is a linear transformation.

For example, let we assume there are three ancestral populations and seven individuals, in which individuals 1, 2, 3 are inherited from the ancestral populations without admixture, individuals 4, 5, 6 have two ancestral populations with equal contributions and individual 7 has three ancestral populations with equal contributions. The AF matrix $\mathbf{R}$ is

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 1/2 & 1/2 & 0 & 1/3 \\ 0 & 1 & 0 & 1/2 & 0 & 1/2 & 1/3 \\ 0 & 0 & 1 & 0 & 1/2 & 1/2 & 1/3 \end{bmatrix}^T,$$

and Θ_A is assumed to $\text{diag}(0.05, 0.05, 0.05)$. The AF coordinates are shown in Fig-

ure 3.4a, and the new eigen-decomposition coordinates are shown in Figure 3.4b.

3.3.4 Proof of Eigen-decomposition

Here, we perform eigen-decomposition on $\Theta_{\mathbf{R}} = (\mathbf{R} - \frac{1}{n}\mathbf{1}_{nn}\mathbf{R})\Theta_A(\mathbf{R} - \frac{1}{n}\mathbf{1}_{nn}\mathbf{R})^T$ in Equation 3.10, and find the mapping between the eigenvectors of $\Theta_{\mathbf{R}}$ and $\mathbf{R}$: it is a linear transformation. Let $Y = \mathbf{R} - \frac{1}{n}\mathbf{1}_{nn}\mathbf{R} = (I_n - \frac{1}{n}\mathbf{1}_{nn})\mathbf{R}$, where I_n is an identity matrix and $\mathbf{1}_{nn}$ is a matrix $n \times n$ with entries equal to one, then $\Theta_{\mathbf{R}} = Y\Theta_A Y^T$.

The facts

1. The singular value decomposition of an $m \times n$ real matrix M is a factorization of the form $M = U\Sigma V^T$, where U is an $m \times m$ real orthogonal matrix, Σ is an $m \times n$ rectangular diagonal matrix with nonnegative real numbers on the diagonal, and V^T is an $n \times n$ real orthogonal matrix. The diagonal entries $\Sigma_{i,i}$ of Σ are known as the singular values of M.

2. A real symmetric matrix A can be decomposed as $A = Q\Lambda Q^T$, where Q is an orthogonal matrix, and Λ is a diagonal matrix whose entries are exactly the eigenvalues of A.

3. The singular value decomposition and the eigen-decomposition are closely related. The left singular vectors of M are eigenvectors of MM^T, the right singular vectors of M are eigenvectors of M^TM, and the non-zero singular values of M (found on the diagonal entries of Σ) are the square roots of the non-zero eigenvalues of both MM^T and M^TM.

Proof.

Note that $\Theta_{\mathbf{R}}$ and Θ_A are not necessarily non-negative definite matrices, and their eigenvalues could be negative. To avoid a complex matrix, we perform eigen-decomposition on $\Theta_{\mathbf{R}}^2$, since $\Theta_{\mathbf{R}}^2$ and $\Theta_{\mathbf{R}}$ have the same eigenvectors and the square of eigenvalues of $\Theta_{\mathbf{R}}$ correspond to the eigenvalues of $\Theta_{\mathbf{R}}^2$.

Note that $\text{rank}(Y) \leq N-1$, then $\text{rank}(\Theta_{\mathbf{R}}) \leq N-1$. Let the eigenvalues of $\Theta_{\mathbf{R}}$ be $|v_1| \geq |v_2| \geq ... \geq |v_{N-1}| \geq |v_N| = ... = |v_n| = 0$, and $Q_{(\mathbf{R}),i}$ be the i^{th} eigenvector with respect to v_i. $[Q_{(\mathbf{R}),1}, ..., Q_{(\mathbf{R}),n}]$ forms an orthogonal matrix.

$$\Theta_{\mathbf{R}}^2 = Y\Theta_A Y^T Y\Theta_A Y^T \tag{3.11}$$

We perform singular value decomposition on Y,

$$\text{SVD} : Y = U_Y \Sigma_Y V_Y^T$$

Since $\text{rank}(Y) \leq N-1$, at least one of the singular values of Y is ZERO. Replace Y in Equation 3.11 by $U_Y\Sigma_Y V_Y^T$:

$$\Theta_{\mathbf{R}}^2 = (U_Y\Sigma_Y V_Y^T\Theta_A V_Y)(\Sigma_Y^T\Sigma_Y)(V_Y^T\Theta_A V_Y\Sigma_Y^T U_Y^T)$$

where $\Sigma_Y^T\Sigma_Y$ forms an $N \times N$ diagonal matrix.

Let $Z_Y = U_Y\Sigma_Y V_Y^T\Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}$, where $\Theta_{\mathbf{R}}^2 = Z_Y Z_Y^T$. SVD on $Z_Y = U_Z\Sigma_Z V_Z^T$. Again, at least one of the singular values of Z is ZERO. Since $\Theta_{\mathbf{R}}^2 = Z_Y Z_Y^T = U_Z\Sigma_Z\Sigma_Z^T U_Z^T$, U_Z is the eigenvector matrix of $\Theta_{\mathbf{R}}^2$, i.e., $[Q_{(\mathbf{R}),1}, ..., Q_{(\mathbf{R}),n}] = U_Z$ and the eigenvalue $|v_i|$ is the singular value of Z_Y (non-negative).

Note that

$$\begin{aligned} U_Z\Sigma_Z = Z_Y V_Z &= (U_Y\Sigma_Y V_Y^T)\Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}V_Z \\ &= Y\Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}V_Z \\ &= (I_n - \tfrac{1}{n}\mathbf{1}_{nn})\mathbf{R}\Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}V_Z \end{aligned}$$

or,

$$\underbrace{[Q_{(\mathbf{R}),1},...,Q_{(\mathbf{R}),N}]\text{diag}(|v_1|,...,|v_N|)}_{\text{eigen coordinate}} = (I_n - \tfrac{1}{n}\mathbf{1}_{nn}) \underbrace{\mathbf{R}}_{\text{AF coordinate}} \Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}V_Z \tag{3.12}$$

□

The left hand side of Equation 3.12 is an $n \times N$ matrix where the last column is ZERO since $v_N = 0$, where as the right hand side is the AF matrix times $(I_n - \frac{1}{n}\mathbf{1}_{nn})$ and $\Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}V_Z$. Note that this transform matrix $\Theta_A V_Y(\Sigma_Y^T\Sigma_Y)^{\frac{1}{2}}V_Z$ is a function of $\mathbf{R}$. However, given an AF matrix $\mathbf{R}$, the transform matrix is determined, so each data point (admixture fraction) in $\mathbf{R}$ maps to a new coordinate by a linear transformation: rotation, scaling, translation, etc.

3.3.5 Inferring Allele Admixture Fraction

The mapping in Figure 3.4 suggests an approach to estimate allele admixture fractions using the largest principal components. Let N be the number of ancestral populations, and $S_1, ..., S_N$ be the samples for pseudo-ancestors, as shown in Figure 3.3. Now we look at the largest $(N-1)$ principal components, and identify each location of pseudo-ancestor i in the eigen coordinates, by averaging the locations in the sample S_i. So we have N positions in the eigen coordinates, which corresponds to N independent components in the AF coordinates. Then a linear transformation

can be made to reverse the original mapping. Finally, the principal components of all study individuals are reversed to the AF coordinates by this linear transformation.

Note that there is a bias in the diagonal shown in Equation 3.10. As the total number of study individuals n increases, $\mathcal{E}[\mathbb{M}^{\mathrm{P}}]$ is more similar to the matrix $\frac{4}{1-\theta_T}\Theta_{\mathbf{R}}$ in terms of $\frac{1}{n^2} \parallel \bullet \parallel_F^2$, where $\parallel \bullet \parallel_F$ is Frobenius norm, i.e.,

$$\frac{1}{n^2} \parallel \mathcal{E}[\mathbb{M}^{\mathrm{P}}] - \tfrac{4}{1-\theta_T}\Theta_{\mathbf{R}} \parallel_F^2 = \frac{4}{n^2} \sum_{j=1}^{n} \frac{(1-\theta_j)^2}{(1-\theta_T)^2} \to 0, \text{as } n \to \infty$$

We might expect that eigen-decomposition on $\mathcal{E}[\mathbb{M}^{\mathrm{P}}]$ and $\frac{4}{1-\theta_T}\Theta_{\mathbf{R}}$ could result in similar eigenvectors corresponding to the largest eigenvalues. Here, "similar" means similar relative positions in the eigen coordinates, since numerical calculation does not guarantee that the resulting eigenvectors will have the same absolute positions in the coordinate, e.g., if a vector v is an eigenvector then $-v$ is also the eigenvector according to the same eigenvalue. Also, $\mathbb{M}^{\mathrm{P}}$ is a nonnegative definite matrix, and its eigenvalues are all nonnegative; however, $\Theta_{\mathbf{R}}$ is not necessarily a nonnegative definite matrix. "Largest eigenvalues" refer to the absolute values of eigenvalues in descending order.

To demonstrate the similarity of relative positions in the eigen coordinates of $\mathcal{E}[\mathbb{M}^{\mathrm{P}}]$ and $\frac{4}{1-\theta_T}\Theta_{\mathbf{R}}$, two pseudo-ancestor populations ($N = 2$) and three admixed populations (admixture fractions 25%, 50%, 75%) with equal sample sizes were utilized here. As shown in Table 3.3, as the sample size of each population grows, the bias for estimating the true admixture fraction 25% and 75% declines from 0.0424 to 0.0004. Another example is a spatially continuous admixed population, i.e., individuals with admixture fractions uniformly distributed from 0 to 1. E.g, if $n = 11$ is the total number of study individuals, there are 11 individuals with admixture fractions

Table 3.3: The bias of estimating population admixture fractions in the example of two pseudo-ancestor populations and three admixed populations with equal sample size n_{pop}.

true admixture fraction	0	0.25	0.5	0.75	1
estimated population admixture fraction[1] from $\mathcal{E}[\mathbb{M}^{\text{P}}]$:					
$n_{\text{pop}} = 1$	0	0.20758	0.50000	0.79242	1
$n_{\text{pop}} = 25$	0	0.24849	0.50000	0.75151	1
$n_{\text{pop}} = 50$	0	0.24925	0.50000	0.75075	1
$n_{\text{pop}} = 100$	0	0.24962	0.50000	0.75038	1

[1]: calculated by averaging allele admixture fractions.

Table 3.4: The bias of estimating allele admixture fractions in the example of a spatially continuous admixed population with n individuals in total[1].

# of individuals n	11	51	101	251	501
the maximum bias of estimated admixture fraction from $\mathcal{E}[\mathbb{M}^{\text{P}}]$	0.02270	0.00548	0.00281	0.00114	0.00057

[1]: allele admixture fractions are uniformly distributed from 0 to 1 derived from two ancestral populations.

of 0%, 10%, 20%, ..., 90% and 100%. The maximum bias of the estimated allele admixture fraction is shown in Table 3.4, and it decreases from 0.02270 to 0.00057 as the total number of individuals n increases.

A scheme for bias removal is to define a new genetic correlation matrix, and let us say $\mathbb{M}^* = [m^*_{j,j'}]$:

$$m^*_{j,j'} = \begin{cases} \dfrac{\sum_{l=1}^{L}(g_{jl} - 2\bar{p}_l)^2 - g_{jl}(2 - g_{jl})}{\sum_{l=1}^{L}\bar{p}_l(1-\bar{p}_l)} & , j = j' \\ \\ \dfrac{\sum_{l=1}^{L}(g_{jl} - 2\bar{p}_l)(g_{j'l} - 2\bar{p}_l)}{\sum_{l=1}^{L}\bar{p}_l(1-\bar{p}_l)} & , j \neq j' \end{cases}, \tag{3.13}$$

then $\mathcal{E}[\mathbb{M}^*] = \frac{4}{1-\theta_T}\Theta_{\mathbf{R}}$ without any bias. Weir & Cockerham (1984) and Weir &

Hill (2002) suggest that the simple modification of taking the ratios of the sums over loci of the numerators and denominators instead of averaging the ratios might reduce variance further for rare variants. $\mathbb{M}^*$ is called an unbiased version of the genetic correlation matrix in this study.

To evaluate the estimated AF from PCA, I empirically compared the AF inferred by $\mathbb{M}^{\mathrm{P}}$, $\mathbb{M}^*$ and those by "frappe" [103] and HAPMIX [87]. The results are shown in Section 3.4.3.

3.3.6 Hierarchical Cluster Analysis

Proposed Measure of Individual Dissimilarity

The proposed measure of individual dissimilarity as shown in Equation 3.2 is repeated here, as a starting point of this section:

$$d_{j,j'} = \begin{cases} \dfrac{\sum_{l=1}^{L} g_{jl}(2-g_{jl})}{2\sum_{l=1}^{L} \bar{p}_l(1-\bar{p}_l)} & \text{, if } j = j' \\ \dfrac{\sum_{l=1}^{L} g_{jl}(2-g_{j'l}) + g_{j'l}(2-g_{jl})}{8\sum_{l=1}^{L} \bar{p}_l(1-\bar{p}_l)} & \text{, if } j \neq j' \end{cases}$$

When the total number of individuals n is large,

$$\mathcal{E}[d_{j,j'}] = \begin{cases} \dfrac{1-\theta_j}{1-\theta_T} & \text{, if } \quad j = j' \\ \dfrac{1-\theta_{jj'}}{1-\theta_T} & \text{, if } \quad j \neq j' \end{cases}$$

Therefore, the individual dissimilarity is directly related to coancestry coefficient (or kinship coefficient). The advantages of this estimator are: **1)** invariant, the expected value is invariant to the allele frequency; **2)** a simple function of coancestry coefficient (or kinship coefficient); **3)** it is a moment estimator and suited for large-scale GWAS

Box 3.3.1 Agglomerative Hierarchical Clustering Algorithm

1. Initially each individual j ($1 \leq j \leq n$) is in its own cluster C_j, i.e., the number of individual in each C_j is one.
2. Repeat step 3 until there is only one cluster left.
3. Merge the nearest cluster, say C_k and $C_{k'}$.
4. Output a dendrogram.

data.

Agglomerative Hierarchical Clustering Algorithm

An agglomerative clustering algorithm is conducted using this individual dissimilarity as shown in Box 3.3.1. The dissimilarity between two clusters C_k and $C_{k'}$ is defined as the average dissimilarity between pairs of individuals, one from C_k and the other from $C_{k'}$:

$$d_{C_k,C_{k'}} = \frac{1}{|C_k| \times |C_{k'}|} \sum_{j \in C_k, j' \in C_{k'}} d_{j,j'} \tag{3.14}$$

where $|C_k|$ denotes the number of individuals in the cluster C_k. Although "the nearest clusters" can be defined by minimum distance "$\min_{j \in C_k, j' \in C_{k'}} d_{j,j'}$" (single linkage in cluster analysis) or maximum distance "$\max_{j \in C_k, j' \in C_{k'}} d_{j,j'}$" (complete linkage), the average dissimilarity in Equation 3.14 tends to be more meaningful from a population perspective. In this study, the average dissimilarity is the focus.

The expected value of individual dissimilarity involves a population-wide coefficient θ_T. Estimation of the kinship coefficient from individual dissimilarity requires a reference population (in Figure 3.2), however the definition of reference population

is arbitrary. Therefore, the reference coancestry could be set to be the first split in the dendrogram from the top. The coancestry coefficient between two clusters in the first split is assumed to be ZERO, and the pairwise kinship coefficients are all calculated relative to the first split. That is, d_{base} is assumed to be the baseline in the dendrogram, then

$$\frac{1}{1-\theta_T} = d_{\text{base}}, \text{ or } \theta_T = 1 - \frac{1}{d_{\text{base}}}$$

The kinship coefficient $\theta_{jj'}$ can be obtained from the individual dissimilarity $d_{jj'}$:

$$\theta_{jj'} = 1 - d_{jj'}/d_{\text{base}} \tag{3.15}$$

This idea will be demonstrated in Figure 3.7 and 3.9 of the next section.

For real data, the number of clusters or the number of isolated populations is usually unknown. The hierarchical cluster technique avoids the problem of setting the number of clusters, as it connects individuals to form "clusters" based on their similarity.

In addition, the dendrogram result can be used to determine how many clusters are in the data. Here, I propose a Z score based on permutation of individuals in two clusters to determine whether the algorithm should divide the individuals into two groups or not. The Z score is defined as follows: consider two known clusters C_k and $C_{k'}$, a pool of individuals from C_k and $C_{k'}$ are divided randomly to two clusters with the same sizes of C_k and $C_{k'}$, and let μ_d be the mean of the distance between randomized clusters and s_d be the corresponding standard deviation. Then

$$Z_{C_k,C_{k'}} \stackrel{\text{def}}{=} (d_{C_k,C_{k'}} - \mu_d)/s_d$$

Box 3.3.2 Determining Clusters From a Dendrogram

1. A dendrogram records each combination between two clusters.

2. Repeat step 3 in the order of creating clusters until there is no combination left.

3. Given two clusters, saying C_k and $C_{k'}$,
   if $Z_{C_k,C_{k'}} < Z_{\text{threshold}}$, *then*
   accept the combination of C_k and $C_{k'}$ as one,
   otherwise,
   C_k and $C_{k'}$ are separate clusters, and any combination involved with C_k and $C_{k'}$ will not be considered further;
   if $|C_k| \leq n_{\text{threshold}}$, then the individuals in C_k are regarded as potential outliers;
   if $|C_{k'}| \leq n_{\text{threshold}}$, then the individuals in $C_{k'}$ are regarded as potential outliers;

4. Output the classification of individuals.

The idea is simple: if C_k and $C_{k'}$ are two separate clusters, then $d_{C_k,C_{k'}}$ should be far away from the mean distance. We apply the decision rule using Z scores to a dendrogram "bottom up", as shown in Box 3.3.2. That is, each individual starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy if satisfying a criteria. There are two parameters $Z_{\text{threshold}}$ and $n_{\text{threshold}}$ in the algorithm. $Z_{\text{threshold}}$ is the threshold of Z score, while $n_{\text{threshold}}$ is used to determine the potential outliers. We will suggest appropriate values for these two parameters empirically in Section 3.4.

Comparison with Identity-by-State Dissimilarity

The genetic dissimilarity was used in several studies based on identity-by-state (IBS) owing to its simplicity [37]. Formally, the IBS dissimilarity between individuals

j and j' is defined as

$$d_{j,j'}^{\text{IBS}} = \frac{1}{2L}\sum_{l=1}^{L}|g_{jl} - g_{j'l}| \tag{3.16}$$

where $g_{jl} \in \{0, 1, 2\}$. Thereby, $d_{j,j'}^{\text{IBS}}$ ranges from zero to one. When individuals j and j' are a pair of identical twins or the same person, $d_{j,j'} = 0$.

Let us focus on one SNP locus. Based on genotypic probabilities in Table 3.2, the expected value is

$$\mathcal{E}[|g_{jl} - g_{j'l}|] = (4 - 2\Delta_7 - \Delta_8 - 4\theta_{jj'})p_l(1 - p_l) - 4p_l^2(1 - p_l)^2\Delta_9 \tag{3.17}$$

where p_l is the allele frequency at SNP l, $\{\Delta_7, \Delta_8, \Delta_9\}$ is a set of Jacquard's IBD coefficients for individuals j and j' and $\theta_{jj'}$ is the kinship coefficient for those individuals. The expected value of $|g_{jl} - g_{j'l}|$ is a function of allele frequency and IBD coefficients. Assuming individuals j and j' are both in an non-inbred population, i.e., $\Delta_1 = ... = \Delta_6 = 0$,

$$\mathcal{E}[|g_{jl} - g_{j'l}|] \;\; = \;\; 4p_l(1 - p_l)(1 - 2\theta_{jj'}) - 4p_l^2(1 - p_l)^2\Delta_9 \tag{3.18}$$

The relationship between allele frequency and the expected value of IBS dissimilarity is shown in Figure 3.5 for full-sibling, parent-child, half-sibling and unrelated pairs respectively.

Compared to IBS dissimilarity, the proposed individual dissimilarity has an advantage in the interpretation, since the expected value of individual dissimilarity is not confounded by allele frequencies (see Equation 3.7).

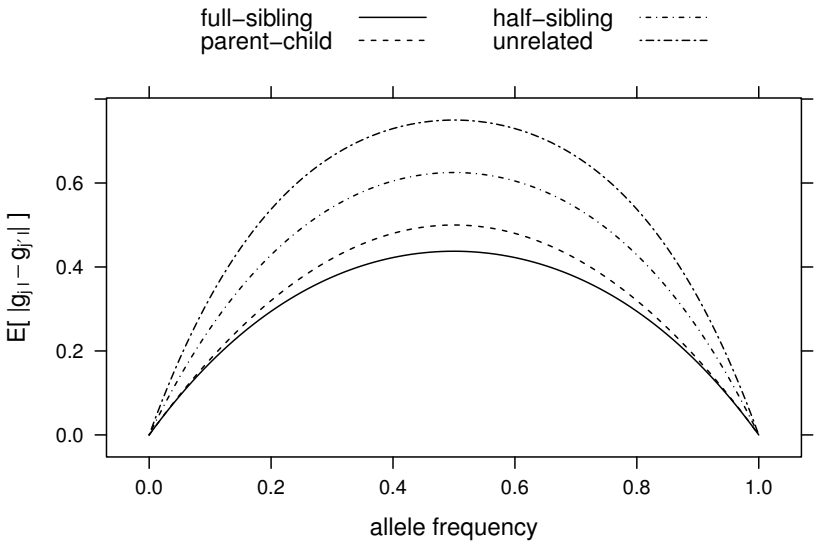


Figure 3.5: The relationship between allele frequency and the expected value of IBS dissimilarity, when one SNP locus l is considered assuming individuals j and j' are in a non-inbred population.

3.4 Application

3.4.1 Materials

In the initial phase (Phase 1 & 2) of the HapMap Project, genetic data were gathered from four populations with European, African and Asian ancestries: CEU, YRI, CHB and JPT, respectively. The genotypic data with PLINK format was downloaded from the website of PLINK: `http://pngu.mgh.harvard.edu/~purcell/plink/res.shtml`. The data consist of 1) 30 parent-offspring trios of Yoruban ancestry from Ibadan in Nigeria, YRI; 2) 30 CEPH trios of European ancestry from Utah, CEU; 3) 45 unrelated Han Chinese from Beijing, CHB; 4) 45 unrelated individuals from Tokyo in Japan, JPT.

The Phase 3 HapMap data consist of SNP genotypes generated from 1,397 sam-

Table 3.5: The population samples in the HapMap project [1].

Name	Population	# in HapMap2	# in HapMap3	# of polymorphic QC+ SNPs[2]
ASW	African ancestry in Southwest USA		87	1543115
CEU	Utah residents with Northern and Western European ancestry from the CEPH collection	90	165	1397814
CHB	Han Chinese in Beijing, China	45	137	1341772
CHD	Chinese in Metropolitan Denver, Colorado		109	1311767
GIH	Gujarati Indians in Houston, Texas		101	1408904
JPT	Japanese in Tokyo, Japan	45	113	1294406
LWK	Luhya in Webuye, Kenya		110	1526783
MEX	Mexican ancestry in Los Angeles, California		86	1453054
MKK	Maasai in Kinyawa, Kenya		184	1532002
TSI	Toscani in Italia		102	1419970
YRI	Yoruba in Ibadan, Nigeria	90	203	1493761

[1]: 1,198 founders and 199 non-founders, 683 males, 714 females.
[2]: with respect to each population in HapMap 3.

ples in total, collected using two platforms: the Illumina Human1M (by the Wellcome Trust Sanger Institute) and the Affymetrix SNP 6.0 (by the Broad Institute). Data from the two platforms have been merged for the release. The PLINK format of HapMap 3 data were downloaded from `http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/hapmap3_r3/plink_format/`. The consensus and polymorphic data set were used in the analyses, which include only SNPs that passed quality control in all populations (i.e., monomorphic SNPs across the entire data set were removed), as shown in Table 3.5.

The methods of PCA and hierarchical cluster analysis were conducted on both the full set (all autosomal SNPs, $n_{\text{snp}} = 1,367,862$) and the pruned set. It is suggested to use a pruned set of SNPs for PCA, which are in approximate linkage equilibrium

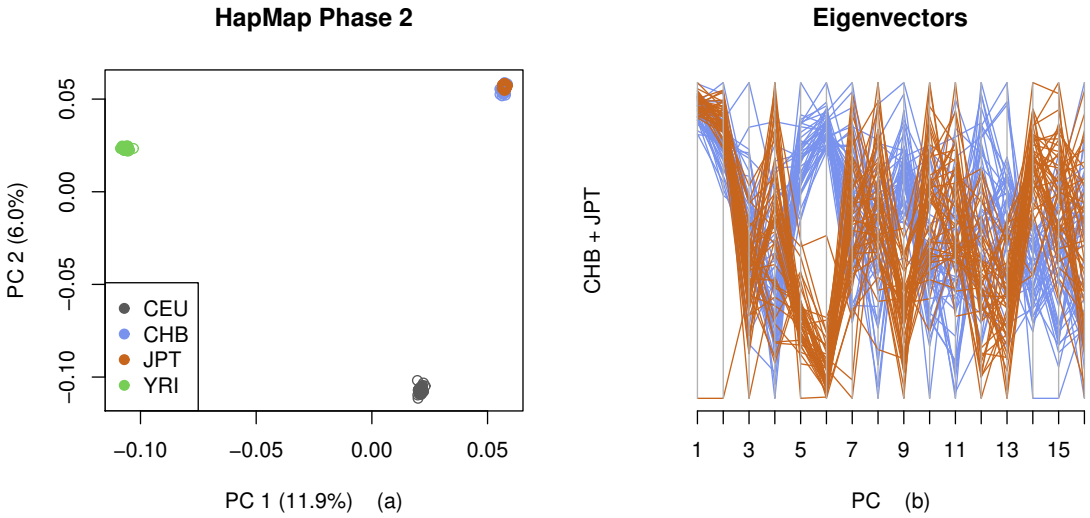


Figure 3.6: The principal component analysis on HapMap Phase 2 data, using the full set of SNP and 210 founders.

with each other to avoid the strong influence of SNP clusters [57]. The R package "SNPRelate" [120] was used to perform linkage disequilibrium pruning to get a subset of SNPs with the default settings, and there were 137,133 SNPs left after LD pruning. In this study, the pruned set of SNPs were used for the sensitivity analysis.

3.4.2 Analyses of HapMap Phase 2 Data

To avoid the confounding effect of related individuals, 210 founders were selected for the PCA analysis removing the offspring from CEU and YRI trio data. The first two principal components are shown in Figure 3.6 (a), and there appear to be three clusters according to different continents. The largest two principal components account for 17.9% (11.9% + 6.0%) of variance in data. Figure 3.6 (b) shows the largest 16 eigenvectors for the combined Chinese and Japanese samples, it seems that the sixth eigenvector can split CHB and JPT samples completely. Although a combination

of multiple eigenvectors may help to improve the identification of population, the choice of eigenvectors is not clear and there will be loss of information when not using all of eigenvectors.

As an alternative to PCA, hierarchical clustering method may have more power to determine the clusters. For the purpose of comparison, the same 210 founders were used in the hierarchical clustering analysis based on individual dissimilarity, as shown in Figure 3.7. The clustering algorithm with Z score was conducted to identify groups that are presented by different ways of shading. The suggested threshold of Z score is 15, which is based on the empirical distribution of Z score for each cluster merging. The corresponding distribution is shown in Figure 3.8. The individuals are labeled by distinct colors with respect to four populations. The identified groups are highly consistent with the populations, except one Japanese who appears more related to Chinese, with this finding in accordance with the discussion made by the communication of the HapMap project. In the paper of HapMap consortium for Phase I project [104], 44 Japanese were used in the report with one removed.

The Y axis in the dendrogram (Figure 3.7) is the dissimilarity between two clusters, referring to Equations 3.7 and 3.14. In detail, the dissimilarity between African and non-African populations is estimated as 1.10, that is the value of one minus the coancestry coefficient $\theta_{\mathrm{A,N}}$ between African and non-African populations over a population-wide factor $1-\theta_T$:

$$\frac{1-\theta_{\mathrm{A,N}}}{1-\theta_T} \quad \text{, or} \quad 1-\frac{\theta_{\mathrm{A,N}}-\theta_T}{1-\theta_T}$$

The latter is one minus relative coancestry coefficient between African and non-African populations (conceptually similar to "$1-\beta_{\mathrm{A,N}}$" in Weir & Hill, 2002). Note that,

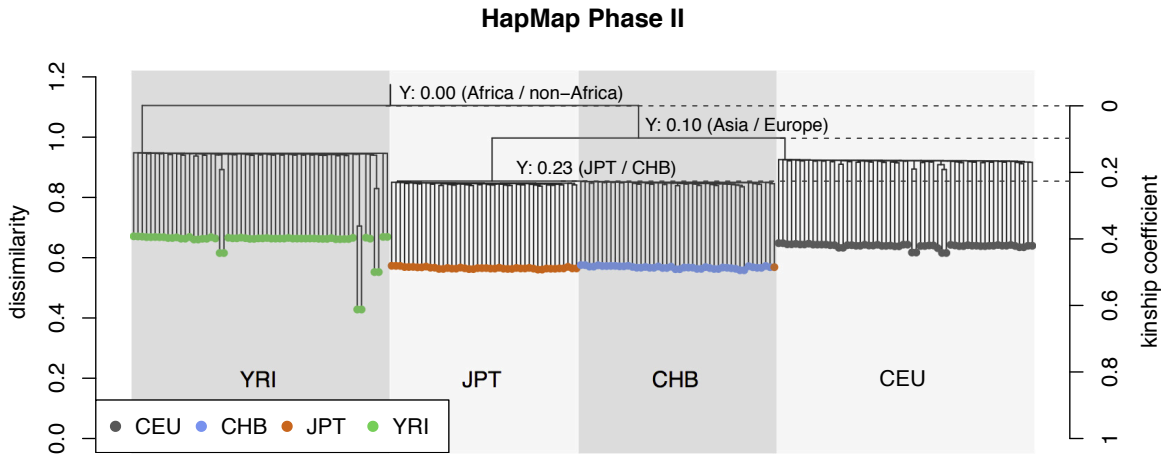
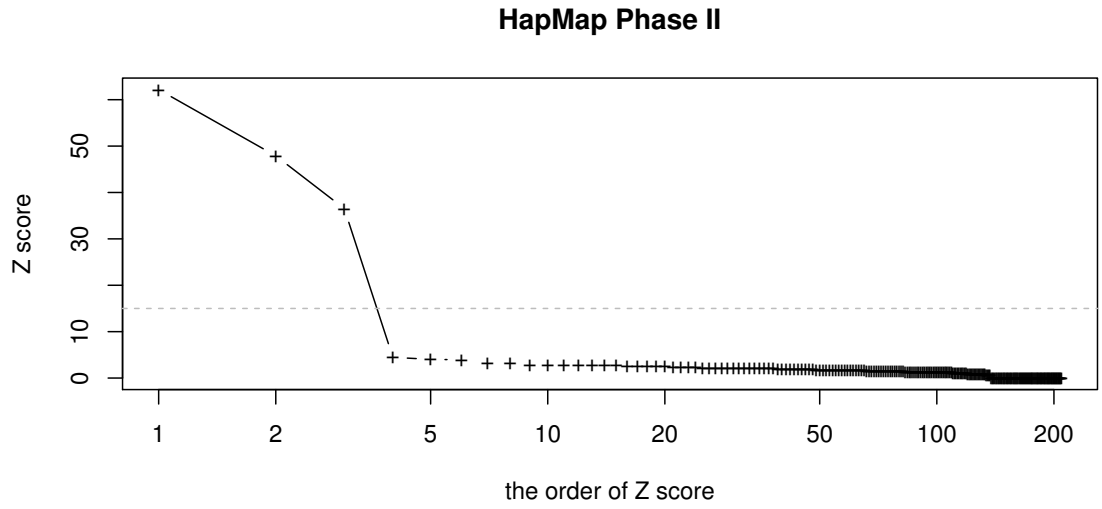


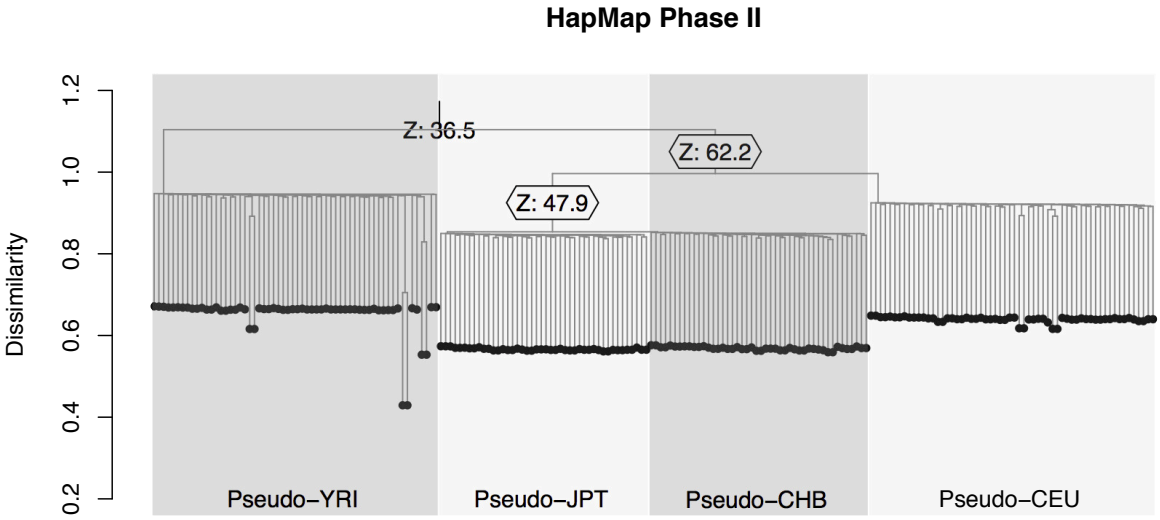
Figure 3.7: The hierarchical clustering analysis on HapMap Phase 2 data, using the full set of SNPs and 210 unrelated founders. Different ways of shading represent the groups identified by the clustering algorithm with the Z score, and thus algorithm found an outlier in the Chinese cluster.

$\theta_{\mathrm{A,N}}$ and θ_T both range from zero to one, hence it is possible to get an estimate of dissimilarity greater than one. An alternative for interpreting the dendrogram is to estimate kinship coefficients by Equation 3.15 with a baseline. The average coancestry coefficient between Asian and European samples is 0.10 relative to the coancestry between African and non-African populations. The inbreeding coefficients and coancestry coefficient within and between CHB and JPT are approximately 0.23 relative to the whole populations. This coancestry-based dendrogram tends to accord with The historical migrations in human evolution: the divergence between Africans and non-Africans occurred before the separation between Asians and Europeans [95].

In addition, the individuals within a population appear to be uniformly similar to each others, except one or two pairs in YRI population. We zoom in Figure 3.7 and focus on the African samples. Figure 3.9 indicates that cryptic relatedness (full-siblings, half-siblings and first cousin) exist in the YRI sample of HapMap Phase 2, which have been reported by previous studies [84].

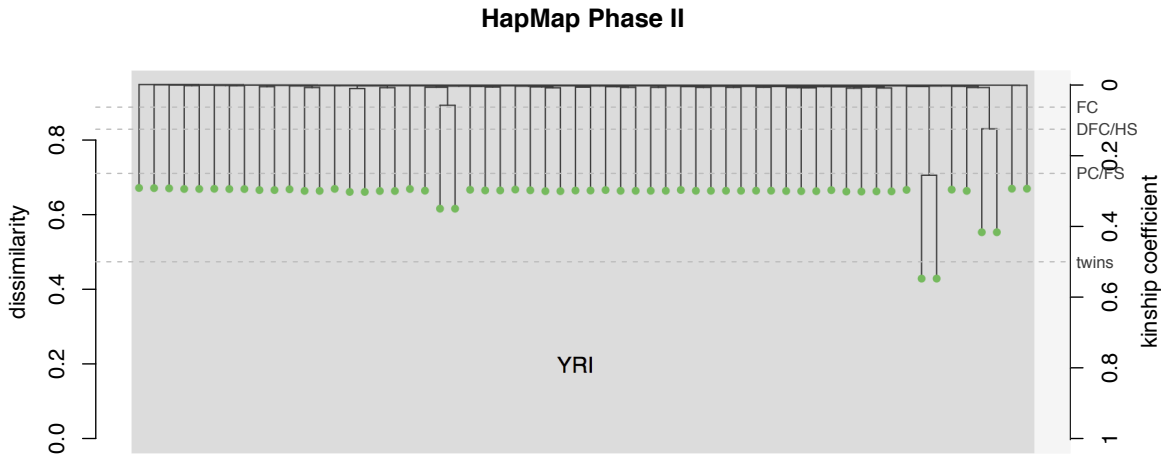


(a)



(b)

Figure 3.8: The empirical distribution of Z score in the clustering algorithm for the HapMap Phase 2 project: a) the first three points correspond to four clusters; b) detailed splits corresponding to Z scores.



PC: parent-child, FS: full siblings, DFC: double first cousin, HS: half siblings, FC: first cousin.

Figure 3.9: The hierarchical clustering analysis on YRI samples of HapMap Phase 2 and kinship estimation, using the full set of SNPs. We zoom in Figure 3.7 and set the baseline to the YRI cluster.

Family data are allowed in the hierarchical clustering with individual dissimilarity, so another hierarchical clustering analysis was conducted using all 270 HapMap individuals. The result is shown in Figure 3.10. For European and African samples, there are two levels of relatedness (see points): "unrelated" trios and parent-offspring pairs. Otherwise, it is quite similar to the result in Figure 3.7. A sensitivity analysis using the pruned set indicates almost the same results, and is not shown here.

3.4.3 Analyses of HapMap Phase 3 Data

To avoid the confounding effect of related individuals, 1,198 founders were selected for the PCA analysis by removing the offspring. The first two principal components are the focus, since more eigenvectors provide little information on inferring primary population structure rather than the first two. As shown in Figure 3.11a, the samples from CEU, YRI and CHB+JPT correspond to three vertices of a triangle, and the

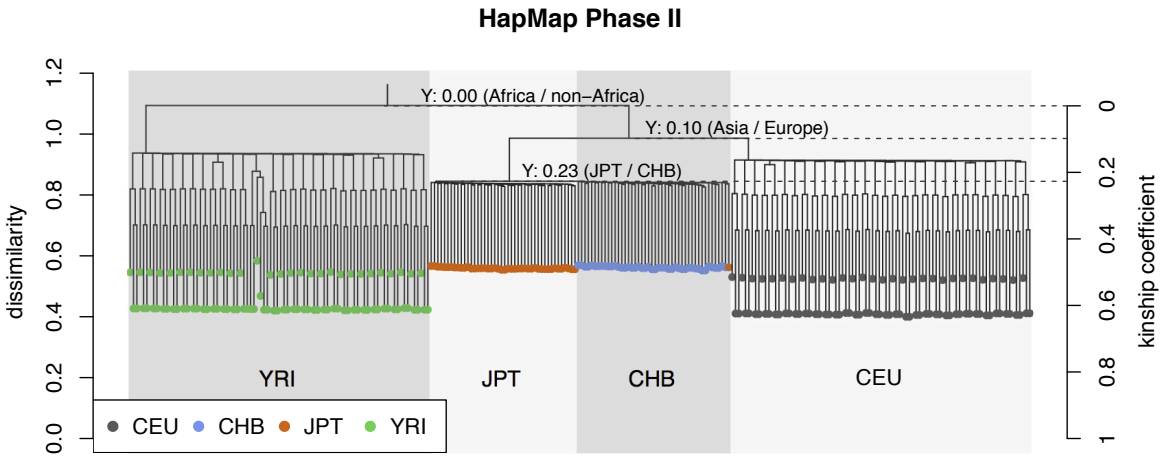


Figure 3.10: The hierarchical clustering analysis on HapMap Phase 2 data, using the full set of SNP and all 270 individuals. Different ways of shading represent the groups identified by the clustering algorithm with Z score, and it founds an outlier in the Chinese cluster.

other populations tend to be admixtures from these three ancestries. This result is pretty similar to what HapMap consortium has done (The International HapMap 3 Consortium, 2010, Figure S2) [50].

Inferring allele admixture fractions was conducted by a coordinate transformation, assuming three ancestral populations and their pseudo-ancestors samples: CEU, YRI and CHB+JPT. The X and Y axes in Figure 3.11b represent the proportions of genomes from African and Asian ancestries respectively. Gujarati Indians in Houston (GIH, yellow) and Mexican ancestry in Los Angeles (MEX, green) appear to be admixtures between Europeans and Asians. ASW, MKK and LWK tend to be more related to African ancestry with some admixture, while CHD and TSI are quite close to the pseudo-ancestors samples.

The population admixture proportions are estimated by averaging allele admixture fractions are shown in Table 3.6, with standard deviation. African Americans (ASW) are a typically admixed sample, estimated with ∼78% of genome from YRI and 21%

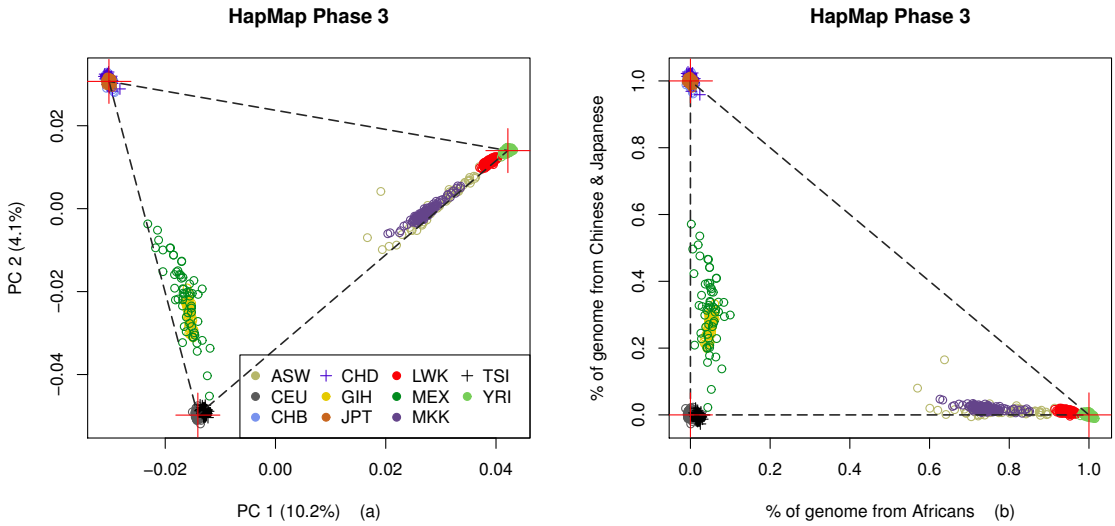


Figure 3.11: The principal component analysis on HapMap Phase 3 data, using the full set of SNP and 1,198 founders consisting of 11 populations: a) the first and second eigenvectors; b) a linear transformation of coordinate from a), assuming three ancestral populations and their pseudo-ancestors samples: CEU, YRI and CHB+JPT. The average positions of three pseudo-ancestors samples are masked by a red plus sign.

from CEU, and approximately no genome from CHB+JPT. The result confirms the estimates of 78% African and 22% European ancestry shown in the supplementary materials (p16) of the HapMap Phase 3 report [50]. The HAPMIX algorithm [87] was used in HapMap Phase 3 project, the optimal linear combination of 74% YRI and 26% CEU was observed for MKK, and a combination of 94% YRI and 6% CEU for LWK. These previous results are quite close to the estimates in Table 3.6. The admixture proportions of GIH and MEX were not reported in the paper of HapMap Phase 3 project, possibly due to lack of appropriate pseudo-ancestor samples. Genetic studies have found the Mexican population to be of mainly Amerindian and European ancestry: 52% European, 44% Native American and 4% African [85]. Note that 4% falls in the 95% confidence interval of inferred YRI fraction for MEX (0.05 ± 0.019).

Table 3.6: The population admixture proportions in the HapMap Phase 3 project [1], estimated by averaging allele admixture proportions of individuals.

population	# of individuals	mean ± sd [2]
ASW	53	(0.78 ± 0.088, 0.01 ± 0.024, 0.21 ± 0.083)
CEU	112	(0.00 ± 0.005, 0.00 ± 0.009, 1.00 ± 0.009)
CHB	137	(0.00 ± 0.004, 1.00 ± 0.011, 0.00 ± 0.010)
CHD	109	(0.00 ± 0.004, 1.01 ± 0.010, -0.01 ± 0.009)
GIH	101	(0.05 ± 0.006, 0.28 ± 0.027, 0.67 ± 0.031)
JPT	113	(0.00 ± 0.004, 1.00 ± 0.007, 0.00 ± 0.007)
LWK	110	(0.94 ± 0.011, 0.01 ± 0.005, 0.05 ± 0.010)
MEX	58	(0.05 ± 0.019, 0.32 ± 0.111, 0.64 ± 0.106)
MKK	156	(0.74 ± 0.035, 0.02 ± 0.007, 0.24 ± 0.032)
TSI	102	(0.02 ± 0.004, 0.00 ± 0.008, 0.99 ± 0.008)
YRI	147	(1.00 ± 0.006, 0.00 ± 0.005, 0.00 ± 0.006)

[1]: inferred by PCA, assuming three ancestral populations and using the full SNP set.
[2]: (% of genome from YRI, % of genome from CHB+JPT, % of CEU)

Note that the linear transformation from Figure 3.11a to 3.11b does not guarantee the estimated AFs ranging from zero to one, but we might expect the average AFs over all study individuals in an admixed population to fall in the range. For example, the genome proportion of CHD from Europeans is labeled as -1%, and it could be interpreted as the departure from the genomic average between CHB and JPT due to genetic variance within a population.

A comparison between PCA and frappe with respect to allele admixture fractions was made using the pruned SNP set. Frappe is a MLE method with an assumption of unlinked markers, therefore the pruned SNP set was used to avoid high linkage disequilibrium. Also, the pseudo-ancestors (YRI, CHB+JPT and CEU) are specified in frappe according to the AFs (1,0,0), (0,1,0) and (0,0,1). As shown in Figure 3.12, the AFs inferred by PCA tend to be consistent with those estimated by frappe using the same SNP set. However, the offsets are observed for two admixed populations

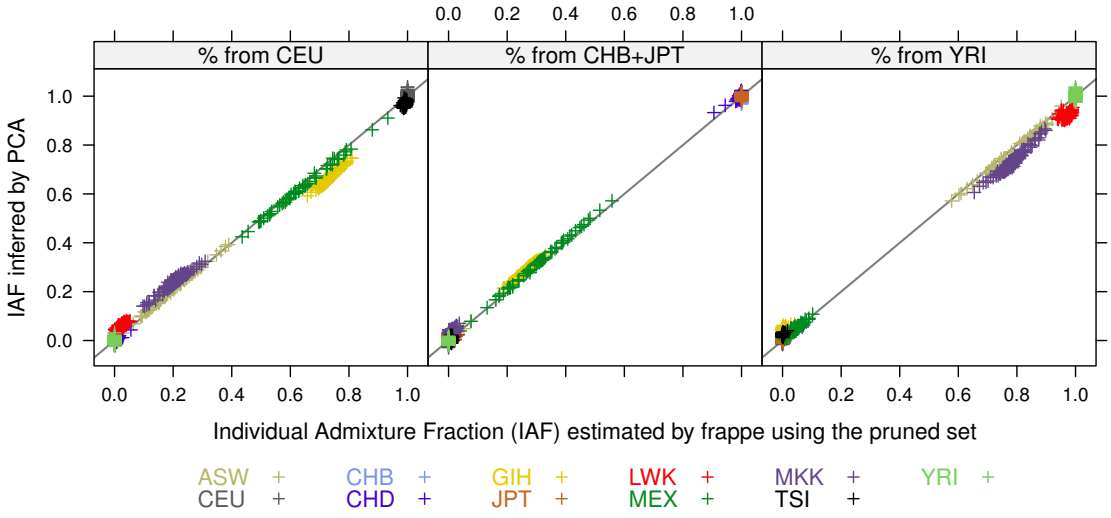


Figure 3.12: A comparison between PCA and frappe with respect to allele admixture fractions. A pruned SNP set ($n_{\text{snp}} = 137,133$) was used by both frappe and PCA.

(GIH and MKK): the PCA-based proportions of genome from CEU are lower than frappe for GIH, and those are higher for MKK. The bias in the diagonal of PCA correlation matrix (see Equation 3.10) is not the reason for these offsets, since an unbiased version of eigen-decomposition ($\mathbb{M}^*$, see Equation 3.13) was also conducted and the offset trend are quite similar to the original PCA (not shown here). Our PCA inference on MKK was actually consistent with what HapMap Phase III has reported. Note that PCA is a dimension reduction technique and may lose information if we look only at the largest two principal components, and the assumption of pseudo-ancestors (CEU, YRI, CHB+JPT) might not truly represent the ancestors in human evolution (see Figure **??**). Meanwhile frappe actually requires unlinked markers and the numerical results could be also subjected to local maximum.

When more SNPs were used, the estimated AFs by the full set are quite close to those by the pruned set (not shown here). However, since AF refers to the proportion

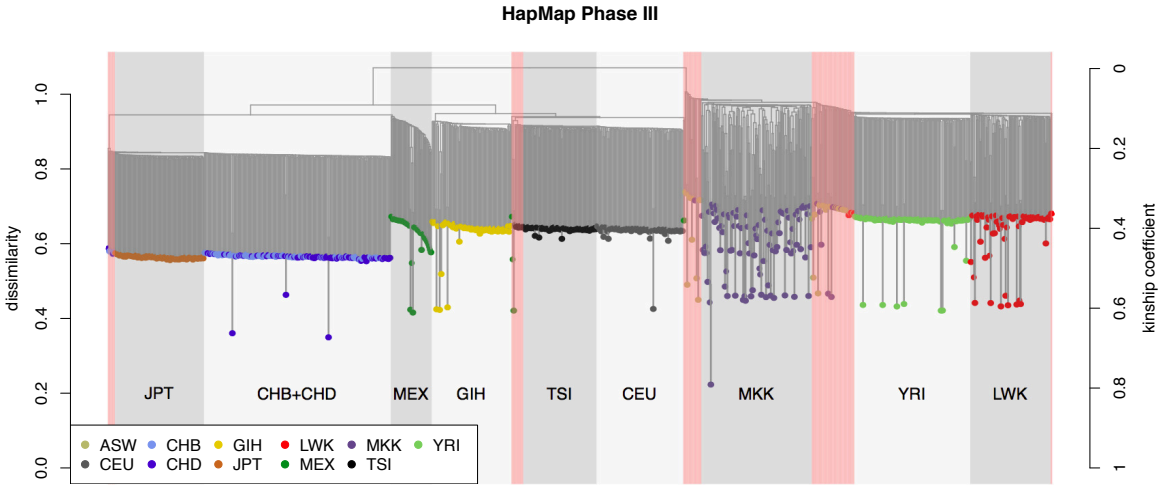
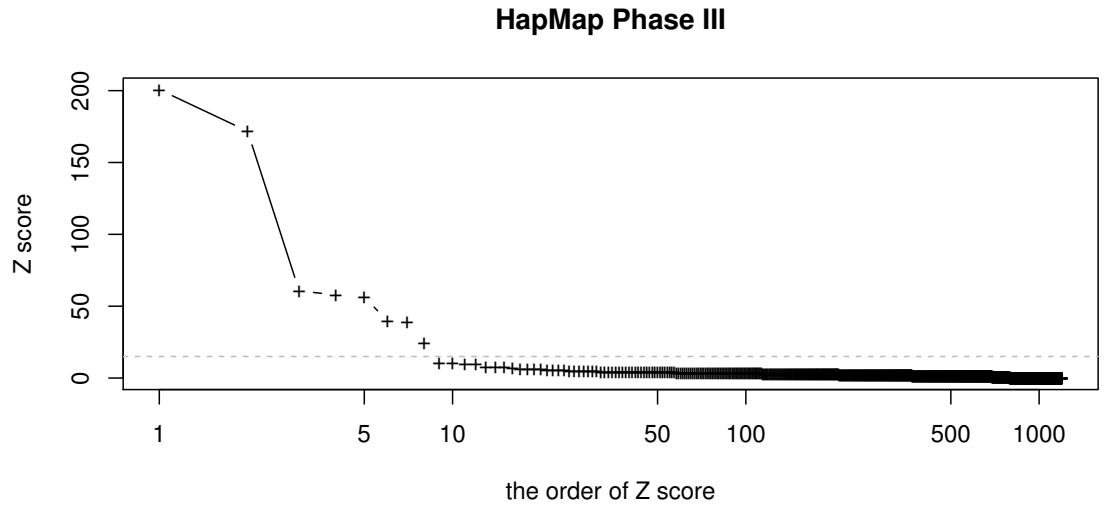


Figure 3.13: The hierarchical clustering analysis on HapMap Phase 3 data, using the full set of SNP and 1,198 founders. Different ways of shading represent the groups identified by the clustering algorithm with Z score, and the pink regions are identified as potential outliers with respect to adjacent clusters.

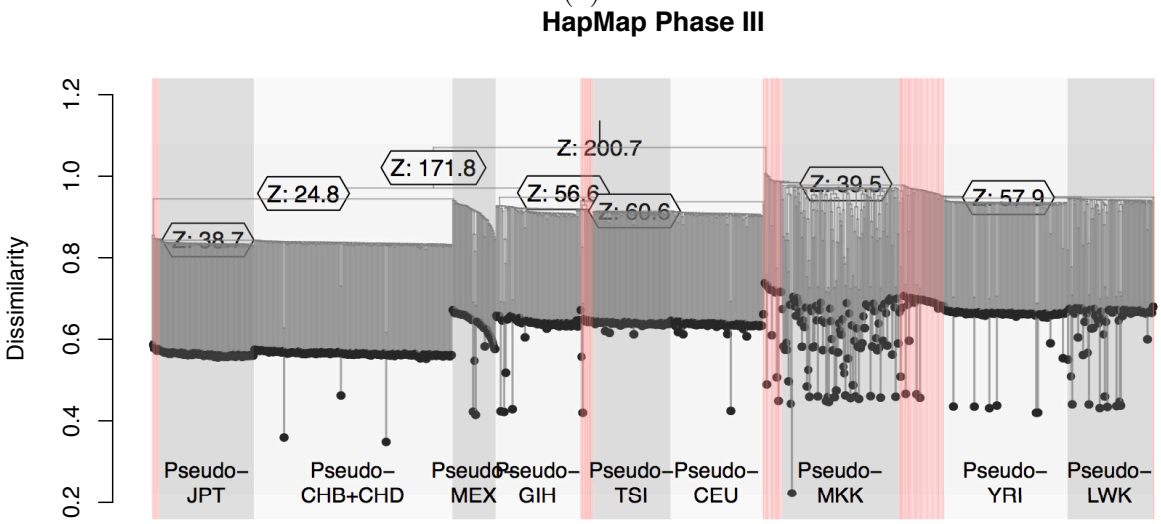
of genome, the true AFs are not necessarily identical for different SNP sets. These results confirm the statistical property derived in Section 3.3.3, and help us understand what PCA does.

As an alternative to PCA, hierarchical clustering method was also applied to 1,198 founders with individual dissimilarity, as shown in Figure 3.13. The clustering algorithm with Z score with the parameter settings: $Z_{\text{threshold}} = 15$ and $n_{\text{threshold}} = 5$. The corresponding distribution is shown in Figure 3.14, and it was conducted to split groups that are presented by different ways of shading, and the pink regions are identified as potential outliers with respect to adjacent clusters.

The individuals are labeled by different colors corresponding to 11 populations. Except for the pink regions, there are 9 clusters left: CHB and CHD are merged together as one group, and ASW samples are distributed around the MKK cluster. The first and second principal components in Figure 3.11 did not split the ASW



(a)



(b)

Figure 3.14: The empirical distribution of Z score in the clustering algorithm for the HapMap Phase 3 project: a) the first eight points correspond to nine clusters; b) detailed splits corresponding to Z scores.

and MKK samples, and both are continually distributed starting from YRI. The dendrogram structure appears to accord with three continents: Europe, Asia and Africa, except Gujarati Indians (GIH) and Mexican ancestry in Los Angeles (MEX).

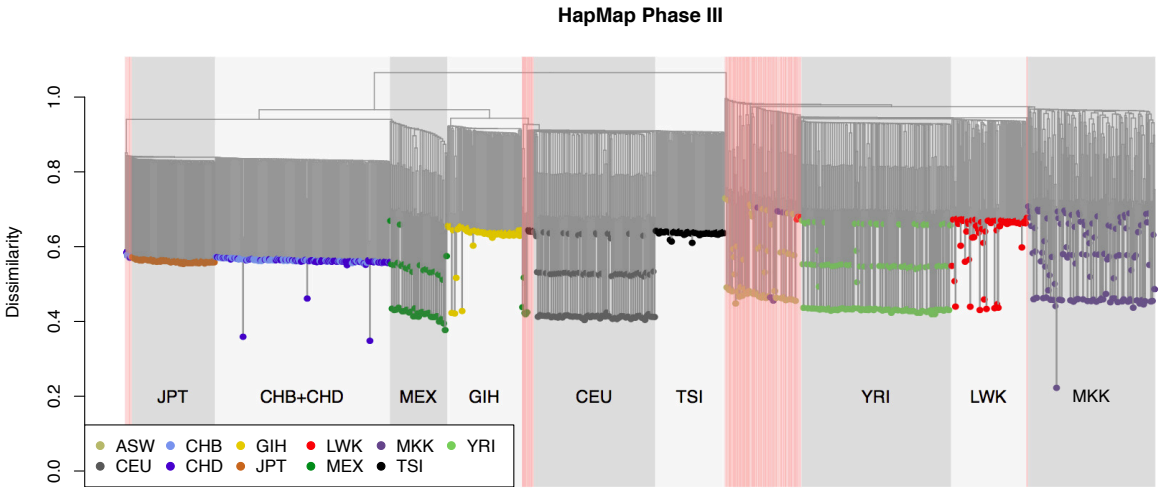


Figure 3.15: The hierarchical clustering analysis on HapMap Phase 3 data, using the full set of SNP and 1,397 individuals. Different ways of shading represent the groups identified by the clustering algorithm with Z score, and the pink regions are identified as potential outliers with respect to adjacent clusters.

Based on the PCA results, GIH and MEX appear more related to Europeans. In addition, another hierarchical clustering analysis was performed using all HapMap3 samples in Figure 3.15, and it is consistent with the previous result.

3.5 Discussion

In this study, two popular data mining tools, principal components analysis (PCA) and hierarchical clustering, were applied to genomic SNP data. I provided interpretations of PCA and hierarchical clustering with a proposed individual dissimilarity from the perspective of identical-by-descent (IBD) probabilities. It is pointed out that a link between largest principal components and allele admixture fractions is an approximately linear transformation, based on individual perspective measures of population structure and assuming multiple ancestral populations shared by admixed samples. Hierarchical clustering analysis split study individuals based on their coancestry coef-

ficients, and successfully separate Chinese and Japanese samples in HapMap Phase II & III data. The main advantage of individual dissimilarity over IBS distance is that it is easy to explain the meaning of the Y axis in a dendrogram and results are not confounded by allele frequency.

It is important to realize the potential limitations and our findings should be interpreted with caution. The assumption of ancestral populations used in inferring admixture fractions from the largest principal components could be confounded by the fact that human evolution is complex and involves with repeated migration and admixture from and out of Africa [20, 3]. Therefore, the selection of pseudo-ancestor samples could be biased due to lack of historical knowledge. For example, it is known that Mexicans mainly are inherited from a mix of Native Americans and European ancestry, with a small African contribution [85]. The admixture proportions of MEX in HapMap Phase 3 data (in Figure 3.6), are confounded by an unknown link between Amerindians and CHB+JPT, although Amerindian seems closely related to Asian rather than European and African in genetics. Also, CHB+JPT and Native Americans represent two evolution branches from their common ancestors, and it may not be appropriate to assume a simple linear combination to reflect genetic difference in Native Americans.

Hierarchical clustering works well on the separation between isolated populations, however it is less efficient to analyze spatially continuous admixed populations. As with African Americans in HapMap Phase 3 data, the clustering algorithm failed to split them from others, since they are related to YRI and LWK gradually. In addition, the potential outliers identified by this algorithm should be checked by other methods to see whether they are true outliers or not. Hierarchical clustering is a greedy heuristic algorithm to find an appropriate tree, but the result is not

necessarily optimal and exhaustive searching is often computationally intractable.

In summary, I provide a genetic interpretation of PCA and propose a hierarchical clustering approach with individual dissimilarity to identity individual clusters. A combination of both two methods should help us better understand population structure for isolated and admixed populations.

Chapter 4

A High-performance Computing Toolset for Big Data Analysis of Genome-Wide Variants

4.1 Abstract

In this big data era, thousands of gigabyte-size data sets are challenging scientists for data management, even on well-equipped hardware. R is one of the most popular statistical programming environments, but it is not typically optimized for high-performance computing necessary for large-scale genome-wide data analyses.

Here I introduce a high-performance C/C++ computing library "CoreArray" [120] for analyses of big-data genome-wide variants. This allows for development of portable and scalable storage technologies, and parallel computing at the multicore and cluster levels. I focus on the application of CoreArray for statisticians working in the R environment. Three R packages gdsfmt, SNPRelate and SeqArray are presented to address or reduce the computational burden associated with the genome-wide association studies.

Gdsfmt provides an R interface for CoreArray that works well generally and outperforms ncdf (v1.6) and rhdf5 (v2.0) on the most of the test datasets I have considered. The benchmarks show uniprocessor implementations of PCA and IBD calculation (defined below) in SNPRelate are ∼10 to 45 times faster than the implementations provided in the popular EIGENSTRAT (v3.0) and PLINK (v1.07) programs,

respectively, and can be sped up to $70 \sim 250$ fold by utilizing eight cores.

SeqArray is designed for data management of sequencing variants, which utilizes the efficient data storage technique and parallel implementation of CoreArray. The 1000 Genomes Project released 39 million genetic variants for 1092 individuals, and a 26G data file was created by SeqArray to store sequencing variants with phasing information, where 2 bits were used as an atomic data type. The file size can be further reduced to 1.3G by compression algorithms without sacrificing access efficiency, since it has a large proportion of rare variants. The uniprocessor benchmark shows that calculating allele frequencies could be done in 5 minutes with the compressed data.

CoreArray will be of great interest to scientists involved in data analyses of large-scale genomic data using R environment, particularly those with limited experience of low-level C programming and parallel computing.

4.2 Background

Today thousands of gigabyte-size data sets are challenging scientists in the management of big data, diverse types of data, and complex data relationships even on well-equipped hardware. In information technology, "big data" usually refers to a collection of data sets so large and complex that it becomes difficult to process them using existing database management tools or traditional data processing applications [75].

Genome-wide association studies (GWAS) have been widely used to investigate the genetic basis of many complex diseases and traits, but the large volumes of data generated from thousands of study samples and millions of genetic variants pose significant computational challenges. In the last ten years, chip-based genotyping technologies, such as the Illumina 1M BeadChip and the Affymetrix 6.0 chip, al-

low hundreds of thousands of common variants (SNPs) across the whole genome to be scored simultaneously. The Gene, Environment Association Studies Consortium (GENEVA) has generated genotypic data using chip-based genotyping techniques, with a large number of research participants ($n > 80,000$) from 14 independently designed studies of various phenotypes [22]. Currently, the field of population genetics is moving from chip data to sequencing data. Next-generation sequencing techniques are being adopted to investigate common and rare variants, making the analyses of large-scale genotypic data even more challenging. For example, the 1000 Genomes Project has identified approximately 38 million single nucleotide polymorphisms (SNPs), 1.4 million short insertions and deletions, and more than 14,000 larger deletions from whole-genome sequencing technologies [2]. In the near future, new technologies, like third-generation whole-genome sequencing [30], will be enabling data to be generated at an unprecedented scale [96]. The computational burden associated with analyses of genome-wide variants is especially evident with large sample and variant sizes, and it requires efficient numerical implementation and memory management.

In this study, the development and application of non-commercial solutions to large-scale GWAS in the big-data era are the focus, although companies such like Google, Microsoft and Amazon have long had mastery of petabyte-scale data sets. The Network Common Data Form (netCDF) and Hierarchical Data Format (HDF) are both popular libraries, designed to store and organize large amounts of numerical data, and they are supported by non-profit research groups. Both libraries were originally written in C, but they also provided interfaces to Fortran, C++ and Java. The netCDF project is hosted by the Unidata program at the University Corporation for Atmospheric Research (`http://www.unidata.ucar.edu/software/netcdf/`), and HDF was originally developed at the National Center for Supercom-

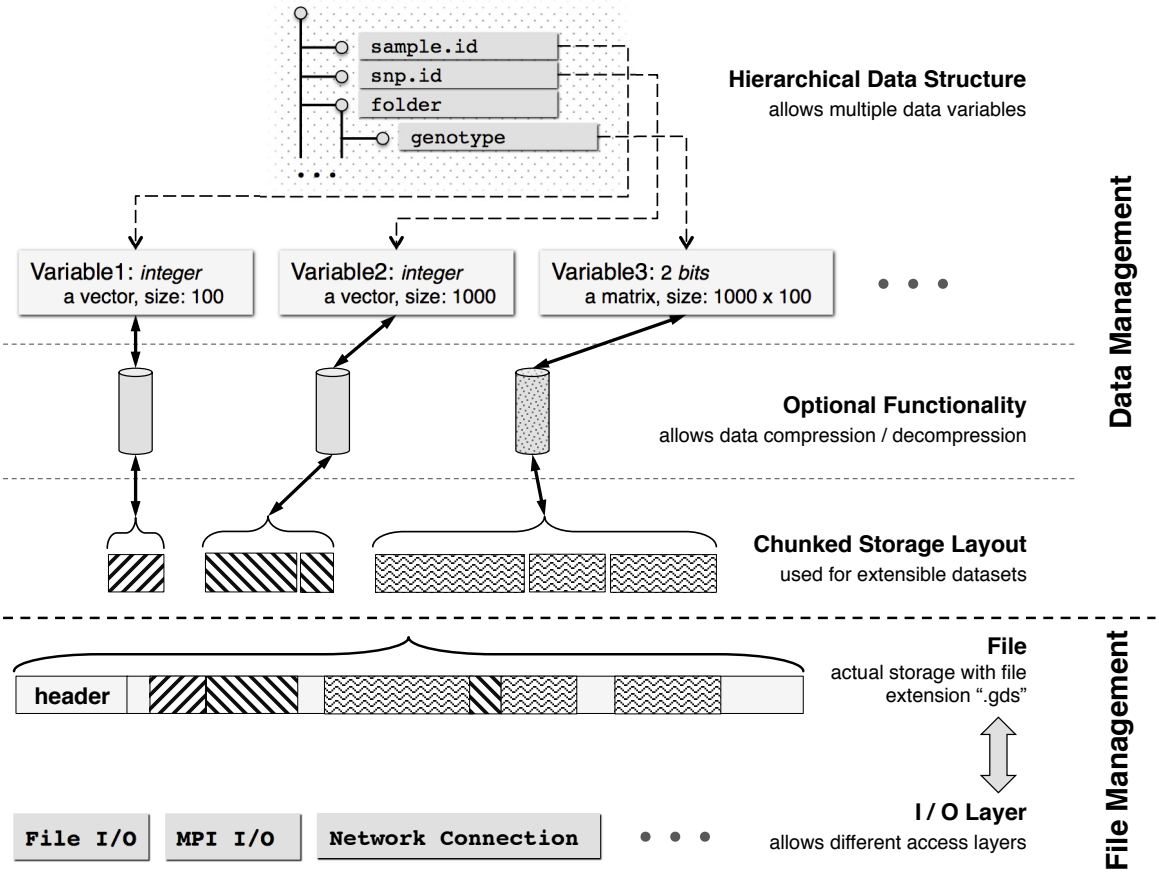
puting Applications and it is currently supported by the non-profit HDF Group (http://www.hdfgroup.org). The latest versions of netCDF and HDF are netCDF-4 and HDF5. Actually, netCDF-4 is based on the kernel of HDF5 and provides a simple high-level application programming interface (API) for HDF5, and it is much more flexible than netCDF-3. The underlying kernel of netCDF-3 is totally different from netCDF-4. NetCDF-3 is the essential format for genotypic data storage in the GENEVA project, and almost all functions in an R/Bioconductor package "GWASTools" [39, 57] associated with GENEVA were originally designed to adopt this format. For purposes of comparison, the performance of netCDF-3 is also demonstrated in this study.

R is one of the most popular statistical programming environments, but it is not typically optimized for high performance or parallel computing which would ease the burden of large-scale GWAS calculations. Direct support of parallel computing in R started with release 2.14.0 (Dec, 2011) including a new package "parallel" shipped with the main program. A CRAN task view for high-performance and parallel computing with R can be found at http://cran.r-project.org/web/views/HighPerformanceComputing.html. For out-of-memory data, two existing R packages "ff" and "bigmemory" offer file-based access to data sets that are too large to be loaded into memory, along with a number of higher-level functions. The bigmemory package was developed by Kane and Emerson, and this project was awarded the 2010 John M. Chambers Statistical Software Award by the ASA Sections on Statistical Computing and Statistical Graphics. However, unlike netCDF and HDF, these two packages do not provide sufficient functions for data management, nor a universal data format to store multiple datasets in a single file.

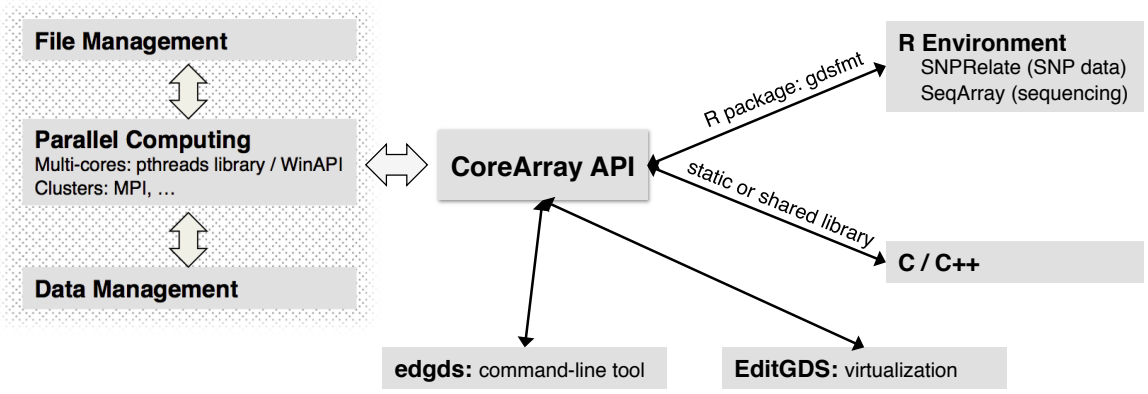
HDF5 is a powerful developer tool for big-data problems and it has been applied

to the fields of Astronomy, Biological and Biomedical Science, Environmental Science and Engineering, etc (`http://www.ncsa.illinois.edu/Projects/`). It is popular in the C/C++, Fortran and Java communities, but not the R community. Currently, an R package allowing limited HDF5 functions is "rhdf5", developed by Fischer from the European Molecular Biology Laboratory in Heidelberg, Germany. SNP data have a special data format, i.e., there are at most 4 cases at a biallelic site which can be stored by 2 bits instead of one byte. There are only 4 single nucleotide A, G, C or T, so it is possible to use less than 8 bits to store that information. In addition, whole-genome sequencing data are not likely to be extremely polymorphic, i.e., there are large proportions of rare variants. Hence the information on variants could be highly compressed, reducing file size and increasing access efficiency of data. HDF5 supports bit-type data via the n-bit filter, but the current version of rhdf5 (v2.0.2) does not provide those functions.

To overcome these limitations and embrace the age of big data, in 2007 I initiated a project named CoreArray (`http://corearray.sourceforge.net/`, hosted by SourceForge that acts as a centralized location for software developers to control and manage free and open source software development). CoreArray was designed for developing portable and scalable storage technologies for bioinformatics data, allowing parallel computing at the multicore and cluster levels. The CoreArray kernel was written in C/C++, but its application is not limited to the C/C++ language. The CoreArray project provides the genomic data structure (GDS) file format for array-oriented data: this is a universal data format to store multiple data variables in a single file. The CoreArray library modules are demonstrated in Figure 4.1 (a) and (b). A hierarchical data structure is used to store multiple extensible data variables in the GDS format, and all datasets are stored in a single file with chunked storage



(a) Data modules.



(b) Application modules (API – Application Programming Interface).

Figure 4.1: CoreArray library modules.

layout. Users can use the CoreArray application programming interface (API) to conduct the functions of file management, data management and parallel computing. The application of CoreArray includes R packages, virtualization and command-line tools.

In this study, I focus on the application of CoreArray for statisticians working in the R environment but with limited C programming experience. CoreArray is less likely to challenge HDF5 in managing primitive data types (e.g., 4-byte integer, floating-point number and characters) by low-level programming in C, but exploiting high-performance API of HDF5 really requires comprehensive C programming knowledges. Here, I provide an efficient R interface "gdsfmt" for creating and access of array-based data. Compared to other R interfaces ("ncdf" to netCDF-3 and "rhdf5" to HDF5), gdsfmt works well generally, and even outperforms "ncdf" on the test datasets used in this study.

The GDS format has been adopted by two research groups: it was used in the PCA and IBD analyses in the GENEVA project; and it was used by the Department of Bioinformatics and Computational Biology, Genentech Inc., for storage of SNP and intensity data.

4.2.1 SNP Data in Genome-wide Association Studies

PLINK, an open-source C/C++ tool, was developed to address the computational challenges for whole-genome association and population-based linkage analyses. Since a biallelic SNP site has at most two different alleles, constituting three possible genotypes with an additional state to indicate missing data. This information can be packed into two bits instead of using one byte, and PLINK reduces the file size by packing 4 SNP genotypes into one byte. The main limitation of PLINK is memory

because PLINK has to load all SNP genotypes to memory. Therefore, the GDS format provides a big-data solution to storing SNP genotypes for GWAS, by alowing access to data as needed without loading all data to memory.

Overall, two R packages have been presented to address some of computational challenges in GWAS: "gdsfmt" to provide efficient, platform-independent memory and file management for genome-wide numerical data, and "SNPRelate" to solve large-scale, numerically-intensive GWAS calculations (i.e., PCA and IBD, see below) on multi-core symmetric multi-processing (SMP) computer architectures [120]. Future development based on the GDS format is allowed, and users could exploit the parallel computing functions in the gdsfmt package with the R package "parallel" to speed up the analyses.

Principal component analysis (PCA) has been proposed to detect and correct for population structure in genetic association studies. The eigen-analysis has been implemented in the software package "EIGENSTRAT" but the computational burden is evident for large scale GWAS SNP data with several thousand study individuals. Parallel computing was formally supported by EIGENSTRAT v4.0, but it still required keeping all data in memory. The PCA functions in the R package SNPRelate allow much larger datasets than does EIGENSTART: the kernel of SNPRelate was written in C and has been highly optimized and it runs faster than EIGENSTART.

Identity-by-descent (IBD) methods have also used to correct for population structure and cryptic relatedness by estimating the degree of relatedness between each pair of study samples [21]. Maximum-likelihood estimation (MLE) was first proposed by Thompson (1975) [105] to estimate three IBD coefficients in a non-inbred population by a "hill climbing" technique. An expectation–maximization (EM) algorithm was proposed by Choi et al. (2009) [21] to estimate IBD coefficients but this is very

time-consuming and not suitable for large-scale data. An alternative is the method of moments (MoM) approach provided by PLINK based on identity-by-state. Compared to MLE, MoM has a great advantage in computational efficiency. The R packages "CrypticIBDcheck" and "ibdreg", and many other binary software written in other languages, involved with IBD coefficients, have a limitation in data scale. By contrast, gdsfmt and SNPRelate provide an efficient data storage technique and a parallel implementation to address and reduce the burden of IBD calculation. Since big-data analyses are the focus of this study, requiring computationally efficient methods, the performance of different implementations of MoM, rather than MLE, are compared here.

4.2.2 Sequencing Variants

The Variant Call Format (VCF) was developed for the 1000 Genomes Project. It is a generic text format for storing DNA polymorphism data such as SNPs, insertions, deletions and structural variants, together with rich annotations (`http://vcftools.sourceforge.net`) [23]. It is most likely stored in a compressed manner with indices for fast data retrieval of variants. A less-flexible binary format (Binary Call Format, BCF) is designed for efficient storing and parsing of VCF records. PLINK/SEQ, a toolset for working with sequencing data, is designed to be complementary to the existing PLINK package (`http://atgu.mgh.harvard.edu/plinkseq/`). PLINK/SEQ was written in C/C++, but it also provides an R interface "Rplinkseq" with limited functions. An R/Bioconductor package "VariantAnnotation" was designed for annotating and filtering genetic variants using VCF files, but is not used for data analysis.

Most of the existing software and packages for the analyses of sequencing data are

not designed for R users, and it is thought that the implementation of the R language is slow and not likely to be suitable for big-data analyses. To address this limitation I developed an R package "SeqArray" that utilizes the efficient data storage technique and parallel implementation of CoreArray. SeqArray provides an alternative data storage to VCF for exploiting genetic variant data, and the kernel of SeqArray is written in C/C++ to speed up the intensive computation for large-scale sequencing data. The primary functions in SeqArray are related to data management, offering efficient access of genetic variants using the R language. It is a solution to make up the gap in data analyses between R users and high-throughput sequencing data. R users can use their own packages to extend the functions and computational efficiency of SeqArray.

The outline of this chapter is described as follows. The features of the CoreArray project are described in Section 4.3 including the C/C++ kernel, the R package SNPRelate for SNP data and the R package SeqArray for sequencing analyses. A performance comparison among three R packages gdsfmt, ncdf and rhdf5 is shown in Section 4.4.1, the advantage in running time is also presented in Section 4.4.2 compared to PLINK and EIGENSTRAT, and the efficiency of SeqArray for sequencing variant data is demonstrated in Section 4.4.3. Finally, a summary is provided in Section 4.5.

4.3 Features

The following features are described in this section: CoreArray modules, the GDS structure for SNP data, and the GDS structure for sequencing data.

Table 4.1: Data types supported by the CoreArray library.

signed integer	
	8-bit integer, 16-bit integer, 24-bit integer, 32-bit integer, 64-bit integer signed integer with 2 bits, signed integer with 3 bits, ..., signed integer with 15 bits
unsigned integer	
	8-bit integer, 16-bit integer, 24-bit integer, 32-bit integer, 64-bit integer unsigned integer with 1 bit, unsigned integer with 2 bits, ..., unsigned integer with 15 bits
floating-point number	
	single-precision number (32 bits), double-precision number (64 bits)
character	
	UTF-8 string, UTF-16 string, UTF-32 string

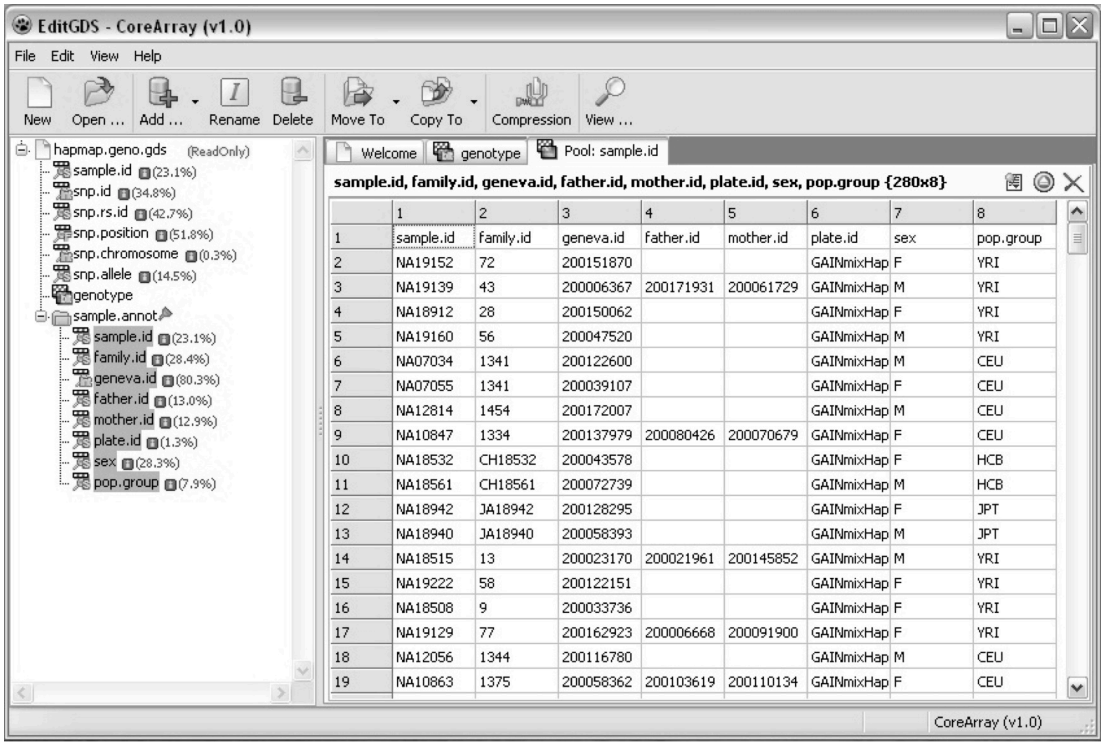
4.3.1 Features of CoreArray

CoreArray is the project name that includes the C/C++ kernel library and external applications. Multiple data variables can be stored in a single file with the universal data format for genomic data structure (GDS). As shown in Table 4.1, the data types are not limited to array-oriented data, and CoreArray also supports storing any file, such as a text file describing project information. Data variables are organized in a hierarchical structure allowing folders to contain different variables. The variable dimension can be extended from any direction. The CoreArray library supports a single file with size of at most 128 terabytes (2^{47} bytes).

The algorithm modules are shown in Figure 4.1a, and the detailed documents for C/C++ source codes are provided at: `http://corearray.sourceforge.net/lib/html/index.html`. Instead of going through the programming details, I provide a higher-level description of CoreArray's functionality here, and an overview of the inheritance diagram for CoreArray object classes on the webpage (`http://corearray.`

sourceforge.net/lib/html/class_core_array_1_1_cd_object.html). For data management, an optional data compression / decompression function can be plugged in. The standard deflate and inflate algorithm, zlib (http://www.zlib.net), is used currently by the CoreArray kernel. A chunked storage layout is adopted in the low-level storage management for extensible datasets, and a contiguous data space may be divided into two or more chunks stored in the actual file without necessarily being adjacent. For example, the user adds 1000 integers to a data variable when a GDS file is created, and then he would like to append another 1000 integers to the variable. However, if the original chunk has no enough space for the new integers, then GDS format will automatically create a new chunk to store the additional data. For file management, CoreArray allows different access I/O layers, such as standard file I/O, MPI I/O (Message Passing Interface, MPI, for parallel computing by multiple processes), etc.

In Unix-like systems, the standard pthreads library is adopted for parallel computing, whereas Windows systems do not provide pthreads by default, instead WinAPIs are called to substitute the functions in pthreads. CoreArray offers a universal platform-independent interface of multi-thread functions. The binary program "EditGDS" was designed for virtualization of GDS format as shown in Figure 4.2. Users can use EditGDS to open a GDS file immediately, and its tree-like structure is automatically displayed in the left panel. Browsing and modifying datasets manually are allowed in EditGDS. EditGDS was written in Free Pascal language (http://www.freepascal.org) using Lazarus (a free development environment). Its source code can be downloaded from http://sourceforge.net/projects/corearray/. A command-line tool "edgds" is shipped with the main C/C++ source codes, allowing multiple operations on a GDS file, such as extracting a subset of the data variables.



	1	2	3	4	5	6	7	8
1	sample.id	family.id	geneva.id	father.id	mother.id	plate.id	sex	pop.group
2	NA19152	72	200151870			GAINmixHap	F	YRI
3	NA19139	43	200006367	200171931	200061729	GAINmixHap	M	YRI
4	NA18912	28	200150062			GAINmixHap	F	YRI
5	NA19160	56	200047520			GAINmixHap	M	YRI
6	NA07034	1341	200122600			GAINmixHap	M	CEU
7	NA07055	1341	200039107			GAINmixHap	F	CEU
8	NA12814	1454	200172007			GAINmixHap	M	CEU
9	NA10847	1334	200137979	200080426	200070679	GAINmixHap	F	CEU
10	NA18532	CH18532	200043578			GAINmixHap	F	HCB
11	NA18561	CH18561	200072739			GAINmixHap	M	HCB
12	NA18942	JA18942	200128295			GAINmixHap	F	JPT
13	NA18940	JA18940	200058393			GAINmixHap	M	JPT
14	NA18515	13	200023170	200021961	200145852	GAINmixHap	M	YRI
15	NA19222	58	200122151			GAINmixHap	F	YRI
16	NA18508	9	200033736			GAINmixHap	F	YRI
17	NA19129	77	200162923	200006668	200091900	GAINmixHap	F	YRI
18	NA12056	1344	200116780			GAINmixHap	M	CEU
19	NA10863	1375	200058362	200103619	200110134	GAINmixHap	F	CEU

Figure 4.2: Virtualization of CoreArray GDS format.

4.3.2 Features of SNPRelate for SNP Data

Data Structure for SNPRelate

To support efficient memory management for genome-wide numerical data, the gdsfmt package provides the genomic data structure (GDS) file format for array-oriented bioinformatic data. This is a container for storing annotation data and SNP genotypes. In this format each byte encodes up to four SNP genotypes, thereby reducing file size and access time. The GDS format supports data blocking so that only the subset of data that is being processed needs to reside in memory. GDS formatted data is also designed for efficient random access to large data sets. Although SNPRelate functions operate only on GDS-format data files, functions to reformat

data from PLINK [91], sequencing Variant Call Format (VCF) [23], netCDF [57] and other data files, are provided by my packages.

```
> # load the R packages: gdsfmt and SNPRelate
> library(gdsfmt)
> library(SNPRelate)
```

Here is a typical GDS file:

```
> snpgdsSummary(snpgdsExampleFileName())

The total number of samples: 279
The total number of SNPs: 9088
SNP genotypes are stored in individual-major mode.
```

snpgdsExampleFileName() returns the file name of a GDS file used as an example in SNPRelate, and it is a subset of data from the HapMap project and the samples were genotyped by the Center for Inherited Disease Research (CIDR) at Johns Hopkins University and the Broad Institute of MIT and Harvard University (Broad). **snpgdsSummary()** summarizes the genotypes stored in the GDS file. "Individual-major mode" indicates listing all SNPs for an individual before listing the SNPs for the next individual, etc. Conversely, "SNP-major mode" indicates listing all individuals for the first SNP before listing all individuals for the second SNP, etc. Sometimes "SNP-major mode" is more computationally efficient than "individual-major model". For example, the calculation of the genetic covariance matrix deals with genotypic data SNP by SNP, and then "SNP-major mode" should be more efficient.

```
> # open a GDS file
> (genofile <- openfn.gds(snpgdsExampleFileName()))
```

```
file name: Users/ZhengX/extdata/hapmap_geno.gds

+         [   ]
|--+ sample.id        [ FStr8 279 ZIP(23.10%) ]
|--+ snp.id        [ Int32 9088 ZIP(34.76%) ]
|--+ snp.rs.id        [ FStr8 9088 ZIP(42.66%) ]
|--+ snp.position        [ Int32 9088 ZIP(51.77%) ]
|--+ snp.chromosome        [ Int32 9088 ZIP(0.33%) ]
|--+ snp.allele        [ FStr8 9088 ZIP(14.45%) ]
|--+ genotype        [ Bit2 9088x279 ] *
|--+ sample.annot        [   ] *
|  |--+ sample.id        [ FStr8 279 ZIP(23.10%) ]
|  |--+ family.id        [ FStr8 279 ZIP(28.37%) ]
|  |--+ geneva.id        [ Int32 279 ZIP(80.29%) ]
|  |--+ father.id        [ FStr8 279 ZIP(12.98%) ]
|  |--+ mother.id        [ FStr8 279 ZIP(12.86%) ]
|  |--+ plate.id        [ FStr8 279 ZIP(1.29%) ]
|  |--+ sex        [ FStr8 279 ZIP(28.32%) ]
|  |--+ pop.group        [ FStr8 279 ZIP(7.89%) ]
```

The output lists all variables stored in the GDS file. At the first level, it stores variables **sample.id**, **snp.id**, etc. The additional information are displayed in the square brackets indicating data type, size, compressed or not + compression ratio. The second-level variables **sex** and **pop.group** are both stored in the folder of **sample.annot**. All of the functions in SNPRelate require a minimum set of variables in the SNP annotation data. The minimum required variables are

- **sample.id**, a unique identifier for each sample.
- **snp.id**, a unique identifier for each SNP.
- **snp.position**, the base position of each SNP on the chromosome, and 0 for unknown position; it does not allow NA.
- **snp.chromosome**, an integer mapping for each chromosome, with values 1-26, mapped in order from 1-22, 23=X,24=XY (the pseudoautosomal region), 25=Y,

26=M (the mitochondrial probes), and 0 for probes with unknown positions; it does not allow NA.

- **genotype**, a SNP genotypic matrix. SNP-major mode: $n_{sample} \times n_{snp}$, individual-major mode: $n_{snp} \times n_{sample}$.

```
> # Take out snp.id
> head(read.gdsn(index.gdsn(genofile, "snp.id")))
[1] 1 2 3 4 5 6

> # Take out snp.rs.id
> head(read.gdsn(index.gdsn(genofile, "snp.rs.id")))
[1] "rs1695824"  "rs13328662" "rs4654497"  "rs10915489" "rs12132314"
[6] "rs12042555"
```

There are two additional variables:

- **snp.rs.id**, a character string for reference SNP ID that may not be unique.
- **snp.allele**, it is not necessary for the analysis, but it is necessary when merging genotypes from different platforms. The format of **snp.allele** is "A allele/B allele", like "T/G" where T is A allele and G is B allele.

There are four possible values stored in the variable **genotype**: 0, 1, 2 and 3. For bi-allelic SNP sites, "0" indicates two B alleles, "1" indicates one A allele and one B allele, "2" indicates two A alleles, and "3" is a missing genotype. For multi-allelic sites, it is a count of the reference allele (3 meaning no call). "Bit2" indicates that each byte encodes up to four SNP genotypes since one byte consists of eight bits. "FStr8" indicates a character-type variable.

```
> # Take out genotype data for the first 3 samples and the first 5 SNPs
> (g <- read.gdsn(index.gdsn(genofile, "genotype"), start=c(1,1), count=c(5,3)))

     [,1] [,2] [,3]
[1,]    2    1    2
[2,]    1    1    1
[3,]    0    0    1
[4,]    1    1    2
[5,]    2    2    2

> # read population information
> pop <- read.gdsn(index.gdsn(genofile, c("sample.annot", "pop.group")))
> table(pop)
pop
CEU HCB JPT YRI
 92  47  47  93

> # close the GDS file
> closefn.gds(genofile)
```

Functions of SNPRelate

SNPRelate provides computationally efficient functions for PCA and IBD relatedness analysis on GDS genotype files. The calculations of the genetic covariance matrix and pairwise IBD coefficients are split into non-overlapping parts and assigned to multiple cores for performance acceleration, as shown in Figure 4.3. The functions in SNPRelate for PCA include the basic calculations of sample and SNP eigenvectors, as well as useful accessory functions. The correlation between sample eigenvectors and observed allelic dosage can be used to evaluate the genome-wide distribution of SNP effects on each eigenvector. The SNP eigenvectors can be used to project a new set of samples to the existing axises, which is useful in studies with substantial relatedness [122].

For relatedness analysis, IBD estimation in SNPRelate can be done by either the method of moments (MoM) [91] or maximum likelihood estimation (MLE) [21, 105]

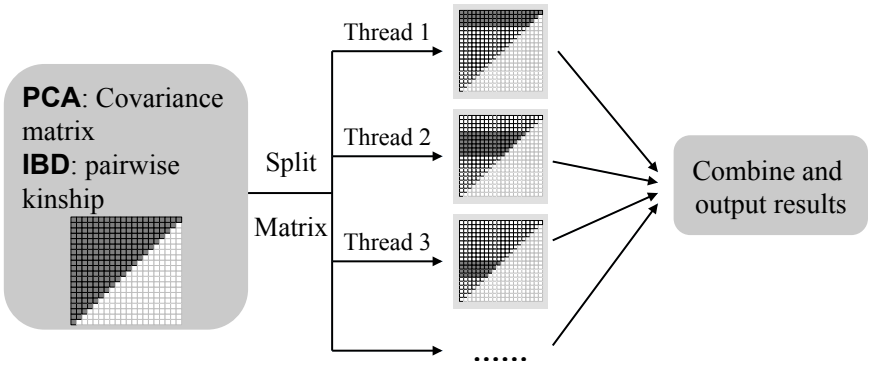


Figure 4.3: Flowchart of parallel computing for principal component analysis and identity-by-descent analysis.

through identity by state (IBS). Our experience shows that MLE is significantly more computationally intensive than MoM for large-scale data analysis, although MLE estimates are usually more reliable than MoM. Additionally, the functions for linkage disequilibrium (LD) pruning generate a pruned subset of SNPs that are in approximate linkage equilibrium with each other, to avoid the strong influence of SNP clusters in PCA and IBD analysis. An actual kinship matrix of individuals can be estimated by either method, which could be used in downstream association analyses [88].

Both R packages are written in C/C++, use the POSIX threads library for shared memory parallel computing on Unix-like systems, and have an R interface in which the kernel has been highly optimized by blocking the computations to exploit the high-speed cache memory. The algorithms are optimized to load genotypes block by block with no limit to the number of SNPs. The algorithms are limited only by the size of the main memory, which is accessed by the parallel threads, and holds either the genetic covariance matrix or IBD coefficient matrix.

GDS is also used by an R/Bioconductor package GWASTools as one of its data storage formats [39]. GWASTools provides many functions for quality control and

analysis of GWAS, including statistics by SNP or scan, batch quality, chromosome anomalies, association tests, etc.

4.3.3 Features of SeqArray for Sequencing Data

A GDS format for storing DNA polymorphism data such as SNPs, insertions, deletions and structural variants, together with rich annotations, including haplotypic phase states, was provided by the R package "SeqArray". A typical GDS for sequencing data with minimal variables is as follows:

```
+
|--+ description     [ ]           <-- indicates sequencing format
|--+ sample.id       [ VStr8, n ]  <-- unique IDs
|--+ variant.id      [ Int32, m ]  <-- unique IDs
|--+ position        [ Int32, m ]  <-- position of the start of the variant
|--+ chromosome      [ VStr8, m ]  <-- chromosome code
|--+ allele          [ VStr8, m ]  <-- reference and alternative alleles
|--+ genotype
|  |--+ data   [ Bit2, (2, n, m1) ]  <-- stores multiple genetic variants
|  |--+ @data  [ Int32, m ]          <-- # of bits needed for each variant / 2
|  |--+ ~data  [ Bit2, (2, m1, n) ]  <-- the transposed "data", optional
|--+ phase
|  |--+ data   [ Bit1, (n, m) ]      <-- 1/0, whether phased or unphased
|  |--+ ~data  [ Bit1, (m, n) ]      <-- the transposed "data", optional
```

where n is the number of samples, m is the total number of variants for DNA polymorphism, (2, n, m1) is a 3-dimensional array, and (n, m) is a matrix, where m1 $\geq$ m. "VStr8" represents variable-length string, whereas "Int32" for 32-bit integer, "Bit2" for 2-bit integer and "Bit1" for 1-bit integer (0/1). The variables **sample.id**, **variant.id**, **position**, **chromosome** and **allele** are not necessarily of data type shown here (VStr8, character; Int32, 32-bit integer). The chromosome code supports "X", "XY", etc, or "Z" for other species. The variable **data** stores multiple genotypic variants in a single 3-dimensional dataset. The size of the first dimension is two,

since human genomes consist of pairs of chromosomes. For other polyploid species, the size of the first dimension could be greater than two to reflect actual number of copies of chromosomes. The type of **genotype/data** is "Bit2" allowing at most four possible values, and it is sufficient to represent most of genetic variants since SNPs are the most common polymorphism (two alleles plus a missing flag, three possible values in total). If a site has more than three possible polymorphisms (like multiple alleles, insertion or deletion), contiguous space will be automatically used to store additional polymorphic information. For example, a site with seven polymorphisms, an allele with eight possible values (seven alleles plus a missing flag) cannot be stored in two bits, then SeqArray will utilize contiguous two bits in the next sub data space to represent this value, i.e., a total of four bits can represent at most 16 classes. Therefore, the third dimension of **genotype/data** `m1` could be greater than `m`. The variable **genotype/length** provides information for how many bits needed for each variant. Finally, **phase** indicates the phasing states have been determined or not by sequencing methods. The chip-based genotyping techniques cannot determine phases or haplotypes, the next-generation sequencing partially offers phasing information but that information is limited in a small DNA fragment. The prefix $\sim$ indicates that it is a transposed version of corresponding variable, which helps for optimizing the access efficiency.

Here I describe only the key features of SeqArray. Additional annotation information, such as quality score, are also able to be stored in the GDS file. Any future extension of SeqArray will depend on real problems, and some coding optimization needs to be made by C programming. Comprehensive R analyses using SeqArray will be provided by other packages from my future research.

Table 4.2: Array variables of 32-bit integer for performance comparisons in R.

Variable Name	Number of Dimensions	Testing	
		Small Dimensions	Large Dimensions
Variable 1	1	[262144]	[10000000]
Variable 2	2	[512][512]	[3162][3162]
Variable 3	3	[64][64][64]	[215][215][215]
Variable 4	4	[22][22][22][22]	[56][56][56][56]
Variable 5	5	[12][12][12][12][12]	[25][25][25][25][25]
Variable 6	6	[8][8][8][8][8][8]	[14][14][14][14][14][14]
Data Size		$\sim$ 1MB	$\sim$ 30 to 40 MB

4.4 Performances

4.4.1 Comparison of R Packages

A benchmark scheme was adopted to make a comparison among three R packages "gdsfmt", "ncdf" and "rhdf5". The array 32-bit integer variables used in the performance comparison are shown in Table 4.2 with dimensions ranging from one to six, which has been used by the HDF Group for a netCDF-4 performance report [i]. The small sets are approximately 1MB, while the sizes of large tests range from 30MB to 40MB. Each benchmark run measured the time to read or write a single variable, and the execution sequence was:

[i] "NetCDF-4 Performance Report – The HDF Group" at `http://www.hdfgroup.org/pubs/papers/2008-06_netcdf4_perf_report.pdf`

```
Repeat 100 times:
    Test 1: Open File1; write Variable 1; close File1
        [ call "drop caches" if cache is disabled ]
    Test 2: Open File1; read Variable 1; close File1
        [ call "drop caches" if cache is disabled ]
    Test 3: Open File2; write Variable 2; close File2
        [ call "drop caches" if cache is disabled ]
    Test 4: Open File2; read Variable 2; close File2
        [ call "drop caches" if cache is disabled ]
    ...
    Test 11: Open File6; write Variable 6; close File6
        [ call "drop caches" if cache is disabled ]
    Test 12: Open File6; read Variable 6; close File6
        [ call "drop caches" if cache is disabled ]
```

The entire benchmark consists of 100 runs, and the read and write speeds are estimated by averaging 100 speeds of each single run. The benchmark uses the command system.time in R to bracket the read and write functions. The read and write speeds are reported in Figure 4.4, calculated by the size of dataset in megabytes over the elapsed wall clock time for the corresponding calls. The latest versions of R packages, gdsfmt (v0.9.10), ncdf (v1.6) and rhdf5 (v2.0.2), were used in the tests. The benchmarks were run on a Linux system with two quad-core Intel processors (2.27GHz) and 32 GB RAM. The system kernel caches programs and data in memory as long as possible, so sometimes read and write rates actually reflect the time to access memory rather than disk. In practice the system caching is almost always enabled, but the file size may be out of the range of cache memory. Therefore, the rates with and without memory cache are both investigated in this study.

Figure 4.4 shows that gdsfmt outperforms the other packages on reading and writing data for every single run when the system cache is enabled. On average, the read rate of gdsfmt is about 3 times of those of ncdf and rhdf5, and the write rate is $\sim$ 1.6 times compared to ncdf, and $\sim$ 14 times faster than rhdf5. Reading and writing on small datasets are slower than the same operations on large datasets. The

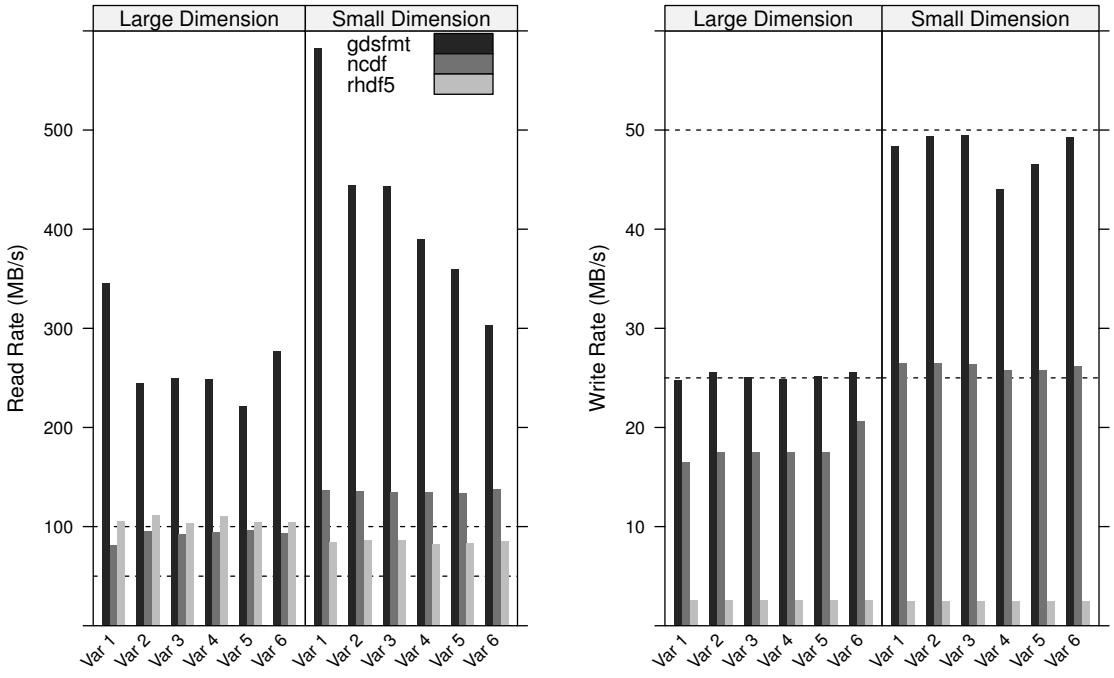


Figure 4.4: The benchmarks for reading and writing data using gdsfmt, ncdf and rhdf5 when the system cache is enabled. Gdsfmt outperforms the other two packages.

read rates of gdsfmt tend to decline as the number of dimensions, whereas the read rates of ncdf and rhdf5 and write rates do not have such trend. When the system cache is cleared (Figure 4.5), the read speed is about twice of the write rate on large datasets for gdsfmt and ncdf packages. Note that, on average, the read rate using system cache is about 5 times of that without system cache. Overall, gdsfmt performs well compared to ncdf and rhdf5.

4.4.2 Comparison with PLINK and EIGENSTRAT

We illustrate the performance of SNPRelate using small, medium and large test data sets. The small and medium sets were constructed from simulated data and

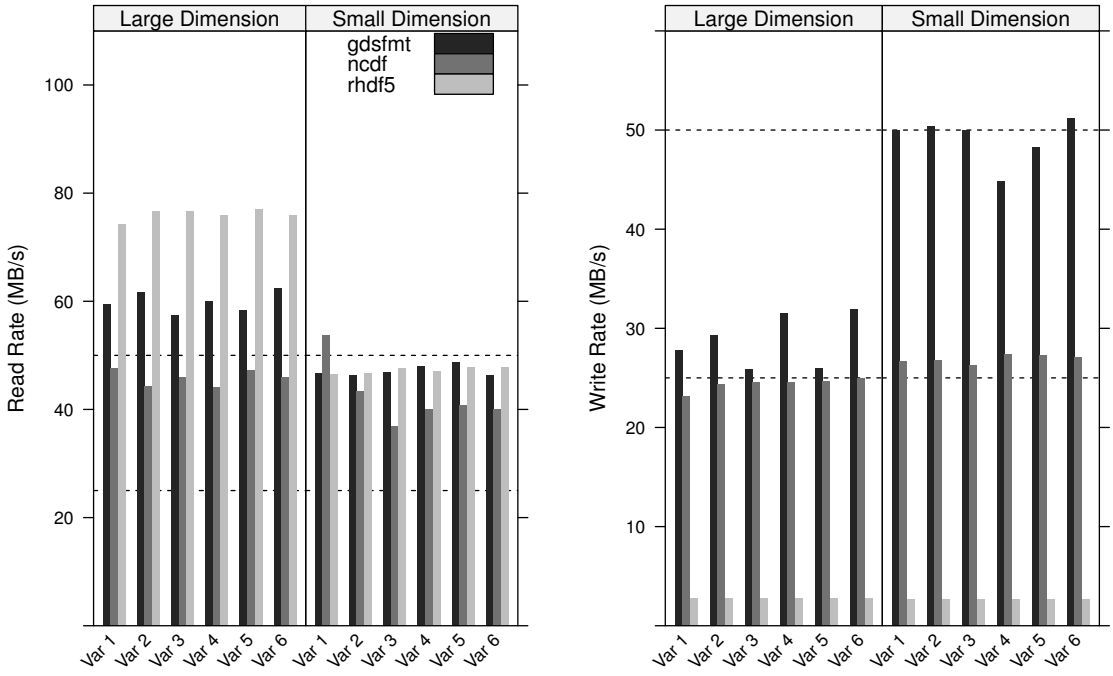


Figure 4.5: The benchmarks for reading and writing data using gdsfmt, ncdf and rhdf5 when the system cache is cleared. Gdsfmt is more efficient than the other two packages on writing data, whereas rhdf5 is the most efficient on reading data.

contain 500 and 5,000 samples with 100K SNP markers, respectively. The large set consists of 55,324 subjects selected from 16 projects of the "Gene-Environment Association Studies" (GENEVA) consortium [22]. We compared the run times of SNPRelate with EIGENSTRAT (v3.0) and PLINK (v1.07) for PCA and IBD estimation respectively. The implementations were benchmarked on a system with two quad-core Intel processors running at 2.27GHz and 32 GB RAM and running Linux Fedora 10.

As shown in Table 4.3, the uniprocessor implementations of PCA and IBD in SNPRelate are approximately eight to 50 times faster than the implementations provided in EIGENSTRAT and PLINK respectively. When the SNPRelate algorithms

Table 4.3: Comparison of run-times (seconds and minutes) for SNPRelate, EIGENSTRAT and PLINK on a Linux system with two quad-core Intel processors (2.27GHz) and 32 GB RAM.

Method / # of cores	Small Set[1] 1	4	8	Medium Set[1] 1	4	8
Principal Component Analysis (PCA)						
SNPRelate	11s+ 1s[2]	5s+ 1s[2]	3s+ 1s[2]	20m+ 12m[2]	8m+ 12m[2]	5m+ 12m[2]
EIGENSTRAT	90s[3]	—	—	710m[3]	—	—
Method of Moment for Identity-by-Descent Analysis (MoM)						
SNPRelate	19s	6s	4s	30m	8m	5m
PLINK	980s	—	—	1630m	—	—

[1]: simulated 500 (small set) and 5000 (medium set) samples with 500K SNPs;
[2]: calls the uniprocessor version of LAPACK in R to compute the eigenvalues and eigenvectors, taking 1s and 12m for the small and medium set respectively;
[3]: includes the computation time of calculating the eigenvalues and eigenvectors.

were run using eight cores, the performance improvement ranged from ∼30 to ∼300. The SNPRelate PCA was conducted on the large data set ($n = 55,324$ subjects with ∼310K selected SNP markers). It took ∼64 hours to compute the genetic covariance matrix (55K-by-55K) when eight cores were used, and ∼9 days to calculate eigenvalues and eigenvectors using the uniprocessor version of the linear algebra package (LAPACK) in R. The analyses on the small- and medium- size data sets required less than 1GB of memory, and PCA on ∼55K subjects required ∼32GB since the genetic covariance matrix is stored in the main memory shared by threads. An improvement on running time for PCA is to employ a multi-threaded version of BLAS to perform the calculation of eigenvalues and eigenvectors instead of the default uniprocessor one. Although SNPRelate is much faster than EIGENSTRAT for PCA or PLINK for IBD estimation using MoM, the results are numerically the same (i.e. identical accuracy).

4.4.3 Performance for Sequencing Variant Data

Currently, the primary application of CoreArray is in the field of bioinformatics. There are only four single nucleotide A, G, C or T, and at most three possible SNP genotypes at a biallelic locus, therefore we can use less than eight bits to represent SNP or sequencing data. In this section, I compare the performance of three different storage schemes to represent genotypic data from the 1000 Genomes Project [2] using GDS format: 1) one byte represents one allele; 2) four SNP alleles are packed in one byte; 3) four SNP alleles are packed in one byte + data compression. Phasing information have be incorporated in the GDS files, since two alleles at a site are stored separately.

The test datasets consist of 39,706,715 variants and 1,092 study samples. The original VCF data files for the 1000 Genomes Project were downloaded from `http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/analysis_results/integrated_call_sets/`. The function "seqVCF2GDS" in the R package SeqArray was used to convert and merge all VCF files, and all ∼39M variants are extracted from the VCF files. I prepared six GDS files ahead which store genotypic data using the above schemes: one byte, two bits and two bits plus compression for sample-by-variant and variant-by-sample storages respectively. "Sample-by-variant" indicates listing all individuals for the first variant before listing all individuals for the second variant etc, whereas "variant-by-sample" indicates listing all variants first.

A benchmark scheme was adopted to evaluate the performance of CoreArray API by R programming, stratified by scan orders: SNP by SNP, or sample by sample. Since genotypes were stored in the GDS files SNP by SNP, the higher speeds of scanning SNP by SNP should be expected. The function "apply.gdsn" in the R package gdsfmt was used to perform scanning, which allows two ways by SNP or by sample. The

Table 4.4: The computing times of CoreArray when reading genotypic GDS files, which consist of 39,706,715 variants and 1,092 study individuals from the 1000 Genomes Project [1].

Running time (minute / hour)		Storage Scheme: one byte (~ 86G)	2 bits (~ 26G)	2 bits + data compression[1]
Store variant by variant:				~ 1.3G (5.2%)
read by variant	1 core	20.4m	5.4m	4.2m
	4 cores	12.3m	1.4m	1.1m
read by sample[2]	1 core	47.8h	56.7m	234.2m
	4 cores	39.6h	14.4m	58.6m
Store sample by sample:				~ 4.8G (19.2%)
read by variant[2]	1 core	98.9m	44.4m	72.8m
	4 cores	36.8m	13.8m	18.9m
read by sample	1 core	28.8m	6.8m	12.6m
	4 cores	16.6m	1.7m	3.2m

[1]: standard "zlib" was used with default settings, compression ratios are 5.2% and 19.2%.
[2]: using the default buffer size 1G. Use of large buffer size can reduce the times of scanning whole dataset, but it should not be out of memory limit.

function "clusterApply.gdsn" was called in the parallel tests for the multi-core system. The functions "apply.gdsn" and "clusterApply.gdsn" are actually coded in low-level C. Scanning means that passing genotypes to a special function but that function did nothing. Each test was replicated 5 times, and the average running time was reported. The benchmarks were run on a Linux system with two quad-core Intel processors (2.27GHz) and 32 GB RAM, and gdsfmt v0.9.12 and SeqArray v0.9.0 were used in the tests.

As shown in Table 4.4, the compression ratio for the sample-by-variant storage scheme is 5.6%, reflecting the fact that whole human genomes are unlikely to be very polymorphic, since the dataset consists of large proportions of variants. According to the variant-by-sample scheme, the compression ratio is 19.2%, which indicates it is less efficient to compress genetic data for a single individual.

The read rates are presented as the running times for scanning the whole dataset variant by variant, or sample by sample. When the reading order agrees with the storage order (i.e., reading by variant according to storing by variant), it is significantly efficient than the case of disagreement. E.g., it only took 4.2 minutes to read the whole data variant by variant, when genotypic data are stored by variant. The storage scheme of "two bits" is the optimal way to store genotypic data. For scanning by variant, the read speed for "two bits" with and without compression are much faster than "one byte" ($\sim$ four times). It can be explained as the size of two-bit GDS file ($\sim$ 26G and 1.3G) can be cached in system memory (32G), but the GDS file for "one byte" ($\sim$ 86G) had exceeded the memory limit. CoreArray kernel has to refresh file caches by loading more data from the hard disk (please compare the performance of gdsfmt with and without caches in Figure 4.4 and 4.5). The read bottleneck also influenced the performance of parallel computing, and the ratio for "one byte" (20.4m/12.3m) is far away from the factor four, compared to the ratios for "two bits" (5.4m/1.4m) and "two bits plus compression" (4.2m/1.1m). When reading data sample by sample according to the sample-by-variant storage scheme, CoreArray kernel automatically adopts a buffer strategy since the variants of a specified sample are not stored contiguously. The corresponding running times are significantly slower than the times of reading by variant in the same column. It is interesting to see how the memory factor influence the running times: I reran all tests on a workstation with 96G memory, the most extreme reading time reduced from 47.8h to 1.2h. System memory cache does have significant effects on the access efficiency.

When the genotypic data are stored sample by sample, reading by sample is more efficient than reading by variant, e.g., 6.8m vs 44.4m, and 12.6m vs 72.8m. Its compression ratio (19.2%) is higher than that (5.2%) compared to the sample-by-variant

storage scheme. Larger file size (4.8G vs 1.3G) indicates more running times (12.6m vs. 4.2m). If the file size is of greater interest, the storage scheme "two bits plus compression" plus "sample-by-variant" appears to be appropriate for normal-equipped hardware, especially for laptops with at most 8G memory. In the case of laptop, $\sim$ 26G of genotypic data are not able to be cached in memory, then scanning such dataset will require more file read operation interacting with hard disk and will be slowed down. The results in Table 4.4 also indicate that the functions reading the "sample-by-variant" dataset sample by sample require an optimized programming skill. For example, a typical function reading by sample is to calculate the missing rate per sample. This function can be revised as a function of reading by variant, then it could be sped up without a parallel scheme. The applications include calculating allele frequencies, and the uniprocessor benchmark shows that calculating allele frequencies could be done in 5 minutes with the compressed data.

4.5 Conclusion

In this study, I introduced a high-performance computing library CoreArray for big-data analyses of genome-wide variants. The CoreArray project was initiated in 2007 with an aim to develop portable and scalable storage technologies for bioinformatics data allowing parallel computing at the multicore and cluster levels. I focus on the application of CoreArray for statisticians working in the R environment but with limited C programming experience. Three R packages gdsfmt, SNPRelate and SeqArray are presented to address or reduce the computational burden associated with the genome-wide association studies.

Gdsfmt provides a general R interface of CoreArray API, and it works well generally, and it even outperforms ncdf and rhdf5 on the most of the test datasets in

this study. The benchmarks show the uniprocessor implementations of PCA and IBD in SNPRelate are ∼10 to 45 times faster than the implementations provided in the popular EIGENSTRAT (v3.0) and PLINK (v1.07) programs respectively, and can be sped up to 70 ∼ 250 fold by utilizing eight cores. SeqArray offers a solution to make up the gap in data analyses between R users and high-throughput sequencing data utilizing parallel computing.

CoreArray will be of great interest to scientists involved in data analyses of large-scale genomic data using R environment, particularly those with limited experience of low-level C programming and parallel computing.

Chapter 5
Summary

5.1 Dissertation Overview

This dissertation consists of three interrelated studies. First, HLA genotype imputation using SNP markers in genome-wide association studies was considered. Genotyping of classical HLA alleles is an essential tool in the analysis of diseases and adverse drug reactions. However, deriving high-resolution HLA types subsequent to whole-genome SNP typing or sequencing is often cost prohibitive for large samples. An attribute bagging (machine learning) approach, HIBAG, was proposed to predict HLA alleles using dense SNP genotypes, taking advantage of the extended haplotype structure within the MHC. The performance of HIBAG was assessed using HLARES data ($n = 2,668$ subjects of European ancestry) as a training set and data from the British 1958 birth cohort study ($n \approx 1,000$ subjects) as independent validation samples. Prediction accuracies for *HLA–A*, *B*, *C*, *DRB1* and *DQB1* range from 92.2% to 98.1%.

Second, it was noted that population stratification can be inferred by a combination of Principal Component Analysis and Hierarchical Cluster Analysis. An interpretation of PCA based on identity-by-descent measures was given, and an approximately linear transformation between the projection of individuals onto principal components and allele admixture fractions assuming two or more ancestral populations was revealed. This explains why the principal component axes often represent

perpendicular gradients in geographic space.

Third, high-performance computing packages for big-data management and analyses of genome-wide variants were described. Currently next-generation sequencing techniques are being adopted to investigate common and rare variants, making the analyses of large-scale genotypic data challenging. A high-performance C/C++ computing library "CoreArray" and associated R packages gdsfmt, SNPRelate and SeqArray were introduced for management and analyses of genome-wide variants. CoreArray will be of great interest to scientists involved in data analyses of large-scale genomic data using the R environment, particularly those with limited experience of low-level C programming and parallel computing.

5.2 Heritability and HLA Imputation

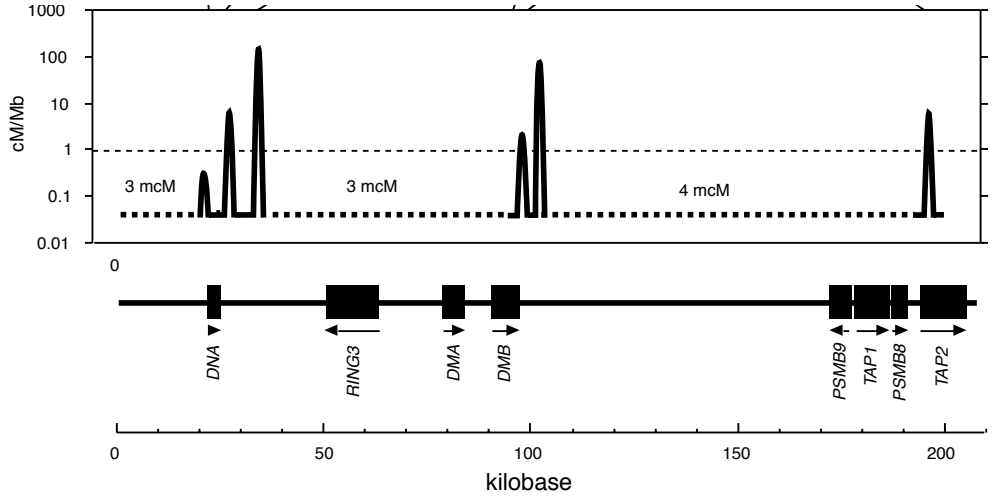
The question of "missing heritability" was mentioned in Chapter One. With respect to common variants in the "common disease – common variant" hypothesis, genetic variation appears to be limited since some alleles at adjacent loci are observed to be correlated, or to be in linkage disequilibrium. Early studies identifying the extent of correlation and structure in haplotypic patterns led to the initiation of the Human Haplotype Map project (HapMap) [7, 50, 104]. LD decays each generation due to chromosome recombination events, and the neutral theory of molecular evolution states that the majority of molecular differences are selectively neutral (not affecting fitness) or nearly neutral [55]. Therefore, the observation that SNP alleles at two close loci are correlated in a population indicates that there were a limited number of recombination events between them and that they have not been affected by selection during their history.

The HLA region has been considered as a high-LD region since it has some of

the largest amounts of correlation between and among loci. However, this high LD does not reflect uniformly low recombination rates within the region, and HLA allelic diversity provides a challenge to evolutionary biologists for an explanation. The high degree of HLA haploytpe diversity seems also to contradict the expected effects of known selection on the region. The most-frequent haplotype may be maintained by epistatic selection of combinations of alleles, or the existence of a mechanism inhibiting recombination in this specific haplotype [107]. These two possible explanations might lead to totally different philosophies of evolution: unlimited variation versus limited variation.

A possible explanation for these observations of high diversity, high LD and low overall recombination lies in the finding of recombination hotspots in the HLA region. As shown in Figure 5.1, there are six recombination hotspots in 216-kb of the HLA region and one of them has a recombination rate of over 100cM/Mb, even though the overall recombination frequency is 0.18 cM over the region. For example, *HLA–B* gene contains 8 exons but its high-resolution HLA typing result refers only to the polymorphism of the first three exons. If meiotic recombination is more likely to occur between the exons 1 and 2 than other regions, then the cumulative effects of recombination over a long population history could create new combinations of exons on a haplotype, even though recombination events in a single meiosis are relatively rare over small regions.

The concept of unlimited variation neither presumes zero correlation between or among loci in the current generation, nor implies that recombination events in a single meiosis should occur much more frequently than has been observed. Instead, it says that any combination of alleles is possible in a large population during a long-term evolutionary stochastic process where dominant characters may be selected. In order



Adapted from Jeffreys et al. (2001) with license number 3163220095447

Figure 5.1: Sperm crossover and local recombination rates in a 216-kb HLA class II region. There were 20,031 single sperm typed from 12 unrelated males, and six hotspots were identified. The total recombination frequency is 0.18 cM over this region (millicentimorgans, mcM, cM $\times 10^{-3}$). The mean rate (0.83 cM/Mb) is shown as a thin dashed line with respect to the 216-kb region, while an estimate of 0.49 cM/Mb across the whole MHC was observed.

to apply this concept, it is necessary to combine both backward-looking (coalescent theory) and forward-looking (classical) population genetic approaches. The alleles shared by all members of a population in the current generation could be tracked back to a single reference population (or the most recent common ancestor) in a coalescent approach, and the genetic variation is limited by this single source. However, starting from a single reference population, evolutionary variation tends to be "unlimited" from a forward perspective. Currently observed alleles could be considered as a sample from a probability space, which starts from the reference population and arrives at the current generation. A coupled (backward and forward) process might promise to fully incorporate natural selection into the theoretical population genetics [110].

HLA imputation can be thought of from the perspective of machine learning. In the field of machine learning, a principal question was posed by Michael Kearns in 1988: can a set of weak learners create a single strong learner? [54]. A weak learner refers to a classifier whose performance is only slightly better than random guessing. The answer is "yes, it is possible". For example, suppose there is a black box known to contain only one of two possible items, an apple or a pear, with equal probability. Suppose it is an apple in the box but this is not known by observers. The first observer makes a prediction by chance: 60% apple and 40% pear, based on his previous experience or knowledge. His decision rule could be considered as a weak classifier since it is just slightly better than random guessing. An ensemble method makes a final prediction by voting from all observers: if over half of them vote "apple" then the decision is "apple," otherwise it is pear. If each observer predicts apple with a probability 60%, and if they (the individual classifiers) are independent from each other, the error rate of the ensemble method goes to zero as the number of classifiers increases. However, if all observers make the same prediction as the first observer, the error rate is still 40%. In other words, te correlation among classifiers will influence the error rate. Therefore, the strength and correlation of individual classifiers will each contribute to the overall error rate. The ensemble method does not always work – it requires that individual classifiers are better than random guessing.

How is the ensemble method related to the HLA imputation problem? Consider two populations **A** and **B** with equal size in the current generation that coalesce to a single population **C** several generations ago. It is assumed that **A** and **B** have evolved under the same evolutionary forces starting from population **C**. The HIBAG training algorithm is applied to population **A**, and population **B** is used as a test set assuming the training model works well in **A**. A combination of backward-looking and forward-

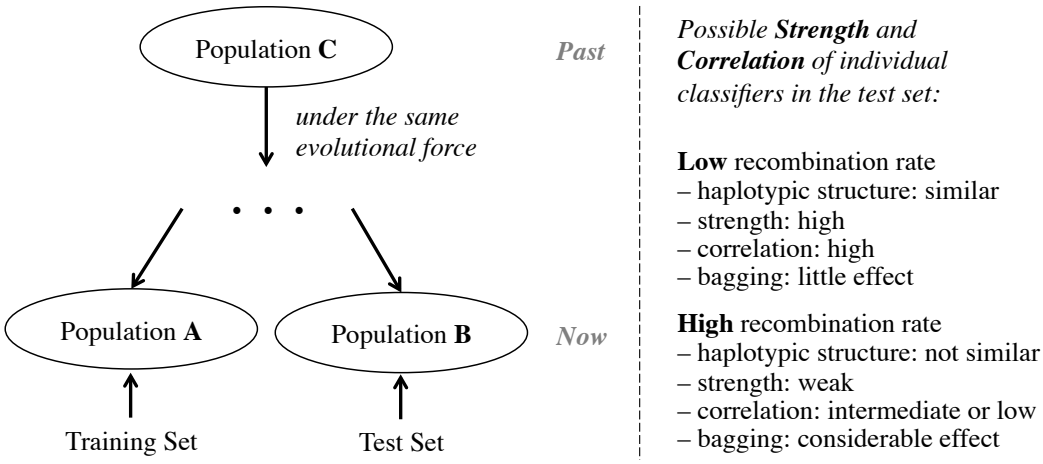


Figure 5.2: The analyses of strength and correlation of individual classifiers in the HIBAG algorithm. Populations **A** and **B** have equal-size, and mild assumptions of evolutionary force are made except for the recombination rate: low mutation rate, no migration, etc. The haplotypic structures of **A** and **B** are compared through $\mathbf{A} \rightarrow \mathbf{C} \rightarrow \mathbf{B}$. The individual classifiers are built on different SNP sets, and this scheme helps to reduce the correlation.

looking perspectives could be applied here. The alleles in **A** are tracked to a reference population **C**, and **B** is created starting from **C** under the same evolutionary forces.

The possible strength and correlation of individual classifiers in the test set are shown in Figure 5.2. In the cases of high recombination rate, the haplotypic structures of **A** and **B** are not likely to be similar, since small changes at the beginning from **C** could result in considerable differences in haplotypes between **A** and **B** after evolution. High rates of recombination might lead to weak individual classifiers with intermediate or low correlation between them, then the ensemble algorithm "bagging" tends to have considerable effect. However, if the recombination rate is too high or coalescence time is too long, the strength of individual classifier is lower than random guessing, and the HIBAG algorithm would fail.

In addition, dissimilar haplotypic structures might be observed in two close pop-

ulations, e.g., Europeans and European Americans. A significant difference in error rate between a single classifier and an ensemble classifier would be expected to be observed. That is one of the reasons for naming the algorithm as "BAG" referring to bagging, an ensemble method. Based on the above logical analyses, it is expected that the HIBAG algorithm could work well even on populations of African ancestry with high genetic diversity, although HLARES data did not fully support this (Table 2.7). It is encouraging that another research group applied HIBAG to their African American samples and found that the 4-digit accuracy of *HLA–DRB1* could attain 90% [60].

5.3 Population Structure

Novembre et al. (2008) showed that SNP profiles of individuals within Europe can be used to infer their geographic origin with surprising accuracy (Figure 5.3) [81]. In principal component analysis, the reason why the principal component axes often represent perpendicular gradients in geographic space, could be explained by allele admixture fractions assuming two or more ancestral populations. In a genetic model (see Figure 3.3), both backward-looking and forward-looking perspectives are considered. The coalescence time according to the single reference population is not specified in the model, thereby it could be many generations ago – even the time before modern humans' ancestors migrated out of Africa. The migration in the history of Europe could create gene frequency clines as suggested by isolation-by-distance models [116], and the surprising accuracy in PCA may indicate high recombination rates across the whole genome. As early as 2004, two studies about recombination hotspots directed by McVean, Donnelly, Myers, et al., have indicated that more than 25,000 hotspots were identified across the human genome using a coalescent-based

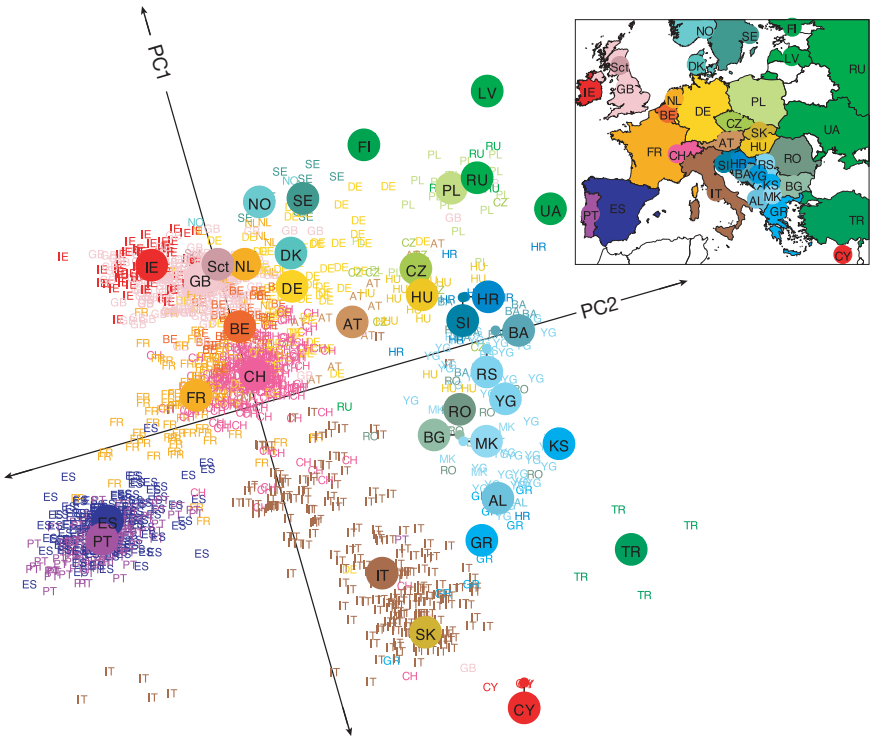
method, although their conclusions were based on small sample sizes and linkage disequilibrium pattern [74, 78].

Starting from the single reference population, such as the population at the time before humans migrated out of Africa, it would be possible to treat the observed alleles and correlation pattern in the current generation as a sample from the probability space of a long-term evolutionary process. This probability space could take the ubiquitous hotspots of the genome into account. The cumulative effects of recombination in hotspots could be considerable if there is no selective pressure, and these effects might suggest a simple strategy of considering linked loci as being weakly correlated or independent. However, it should be noted that even two unlinked loci are not likely to be independent with each other in this probability space if epistatic selection pressure exists. Moreover, this strategy could be confounded by the unknown allele frequencies in the reference population of many generation ago. To avoid this problem, the derivation of the formulas in principal component and hierarchical cluster analyses have taken out the allele frequencies.

The important implications from the studies of HLA imputation and PCA are summarized as: 1) high rates of recombination in the HLA region; 2) high rates of recombination across the whole genome; 3) selective pressure has strong effects on combinations of alleles at multiple loci. High recombination rates in hotspots may also need to be considered in seeking the reason of "missing heritability" in GWAS, since adjacent SNPs may not always have low recombination rates between them.

5.4 Data Challenges

High rates of recombination provide a good reason for next-generation sequencing and then third-generation whole-genome sequencing. The large-scale sequence data



Adapted from Novembre et al. (2008) with license number 3163230279669

Figure 5.3: The relationship between the principal component axes and geographical map within Europe. Individual's SNP profiles can be used to infer their geographic origin with surprising accuracy.

have made the analyses of genotypic variants challenging, and the computational burden becomes the major motivation for developing high-performance computing toolsets for big-data management and analyses of sequencing variants. A solution from the CoreArray project has been proposed and described in Chapter 4.

5.5 Future Directions

The HIBAG classifiers were built based on an intersection of data from several Illumina platforms, and using denser SNP panels did not significantly improve over-

all accuracies. This observation implicitly supports the common disease – common variant hypothesis. Consideration of the infinitesimal model may be a fruitful future direction here. Furthermore, the HLA region contains many structural variations, indicating structural variants might also have important contribution to the increase risk in common disease with SNPs.

The planned next steps are to 1) evaluate and confirm the performance of HIBAG for other ethnic groups, especially for the population of African ancestry because of it is the oldest ancestral population for modern humans; 2) incorporate the estimation of heritability into the HIBAG algorithm to capture epistatic effects across the genome, which requires the development of computing packages; 3) search for deeper principle from applying the HIBAG algorithm to additional datasets.

BIBLIOGRAPHY

[1] 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature 467*, 7319 (Oct 2010), 1061–1073.

[2] 1000 Genomes Project Consortium, Abecasis, G. R., Auton, A., Brooks, L. D., DePristo, M. A., Durbin, R. M., Handsaker, R. E., Kang, H. M., Marth, G. T., and McVean, G. A. An integrated map of genetic variation from 1,092 human genomes. *Nature 491*, 7422 (Nov 2012), 56–65.

[3] Abi-Rached, L., Jobin, M. J., Kulkarni, S., McWhinnie, A., Dalva, K., Gragert, L., Babrzadeh, F., Gharizadeh, B., Luo, M., Plummer, F. A., Kimani, J., Carrington, M., Middleton, D., Rajalingam, R., Beksac, M., Marsh, S. G., Maiers, M., Guethlein, L. A., Tavoularis, S., Little, A. M., Green, R. E., Norman, P. J., and Parham, P. The shaping of modern human immune systems by multiregional admixture with archaic humans. *Science 334*, 6052 (Oct 2011), 89–94.

[4] Alexander, D. H., Novembre, J., and Lange, K. Fast model-based estimation of ancestry in unrelated individuals. *Genome Res 19*, 9 (Sep 2009), 1655–1664.

[5] Anderson, A. D., and Weir, B. S. A maximum-likelihood method for the estimation of pairwise relatedness in structured populations. *Genetics 176*, 1 (May 2007), 421–440.

[6] Balding, D. J., and Nichols, R. A. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica 96*, 1-2 (1995), 3–12.

[7] Barrett, J. C., Fry, B., Maller, J., and Daly, M. J. Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics 21*, 2 (2005), 263–265.

[8] Botstein, D., and Risch, N. Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease. *Nat Genet 33 Suppl* (Mar 2003), 228–237.

[9] Breiman, L. Bagging predictors. In *Machine Learning* (1996), pp. 123–140.

[10] Breiman, L. Heuristics of instability and stabilization in model selection. *The Annals of Statistics 24*, 6 (1996), 2350–2383.

[11] Breiman, L. Out-of-bag estimation. Tech. Rep. ftp.stat.berkeley.edu/pub/users/breiman/OOBestimation.ps, University of Berkeley, 1996b.

[12] Breiman, L. Random forests. *Machine Learning 45* (2001), 5–32. 10.1023/A:1010933404324.

[13] Browning, B. L., and Browning, S. R. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *Am J Hum Genet 84*, 2 (Feb 2009), 210–223.

[14] Browning, S. R. Multilocus association mapping using variable-length Markov chains. *Am J Hum Genet 78*, 6 (Jun 2006), 903–913.

[15] Browning, S. R., and Browning, B. L. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am J Hum Genet 81*, 5 (Nov 2007), 1084–1097.

[16] Browning, S. R., and Browning, B. L. Haplotype phasing: existing methods and new developments. *Nat Rev Genet 12*, 10 (Oct 2011), 703–714.

[17] Bryll, R., Gutierrez-Osuna, R., and Quek, F. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition 36*, 6 (2003), 1291 – 1302.

[18] Buhlmann, P., and Wyner, A. J. Variable length markov chains. *Annals of Statistics 27* (1999), 480–513.

[19] Cardon, L. R., and Palmer, L. J. Population stratification and spurious allelic association. *Lancet 361*, 9357 (Feb 2003), 598–604.

[20] Cavalli-Sforza, L., and Feldman, M. The application of molecular genetic approaches to the study of human evolution. *Nature Genetics 33* (2003), 266–275.

[21] Choi, Y., Wijsman, E. M., and Weir, B. S. Case-control association testing in the presence of unknown relationships. *Genet Epidemiol 33*, 8 (Dec 2009), 668–678.

[22] Cornelis, M. C., Agrawal, A., Cole, J. W., Hansel, N. N., Barnes, K. C., Beaty, T. H., Bennett, S. N., Bierut, L. J., Boerwinkle, E., Doheny, K. F., Feenstra, B., Feingold, E., Fornage, M., Haiman, C. A., Harris, E. L., Hayes, M. G., Heit, J. A., Hu, F. B., Kang, J. H., Laurie, C. C., Ling, H., Manolio, T. A., Marazita, M. L., Mathias, R. A., Mirel, D. B., Paschall, J., Pasquale, L. R., Pugh, E. W., Rice, J. P., Udren, J., van Dam, R. M., Wang, X., Wiggs, J. L., Williams, K., Yu, K., and GENEVA Consortium. The Gene, Environment Association Studies consortium (GENEVA): maximizing the knowledge obtained from GWAS by collaboration across studies of multiple conditions. *Genet Epidemiol 34*, 4 (May 2010), 364–372.

[23] Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., McVean, G., Durbin, R., and 1000 Genomes Project Analysis Group. The variant call format and vcftools. *Bioinformatics 27*, 15 (Aug 2011), 2156–2158.

[24] Dausset, J., Legrand, L., Lepage, V., Contu, L., Marcelli-Barge, A., Wildloecher, I., Benajam, A., Meo, T., and Degos, L. A haplotype study of HLA complex with special reference to the HLA-DR series and to Bf. C2 and glyoxalase I polymorphisms. *Tissue Antigens 12*, 4 (Oct 1978), 297–307.

[25] de Bakker, P. I., McVean, G., Sabeti, P. C., Miretti, M. M., Green, T., Marchini, J., Ke, X., Monsuur, A. J., Whittaker, P., Delgado, M., Morrison, J., Richardson, A., Walsh, E. C., Gao, X., Galver, L., Hart, J., Hafler, D. A., Pericak-Vance, M., Todd, J. A., Daly, M. J., Trowsdale, J., Wijmenga, C., Vyse, T. J., Beck, S., Murray, S. S., Carrington, M., Gregory, S., Deloukas, P., and Rioux, J. D. A high-resolution HLA and SNP haplotype map for disease association studies in the extended human MHC. *Nat Genet 38*, 10 (Oct 2006), 1166–1172.

[26] Delaneau, O., Marchini, J., and Zagury, J. F. A linear complexity phasing method for thousands of genomes. *Nat Methods 9*, 2 (Feb 2012), 179–181.

[27] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*, 1 (1977), 1–38.

[28] Dilthey, A. T., Moutsianas, L., Leslie, S., and McVean, G. HLA*IMP – an integrated framework for imputing classical HLA alleles from SNP genotypes. *Bioinformatics 27*, 7 (Apr 2011), 968–972.

[29] Eichler, E. E., Flint, J., Gibson, G., Kong, A., Leal, S. M., Moore, J. H., and Nadeau, J. H. Missing heritability and strategies for finding the underlying causes of complex disease. *Nat Rev Genet 11*, 6 (Jun 2010), 446–450.

[30] Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., Dewinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden, D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J., and Turner, S. Real-time DNA sequencing from single polymerase molecules. *Science 323*, 5910 (Jan 2009), 133–138.

[31] Eizaguirre, C., Lenz, T. L., Kalbe, M., and Milinski, M. Rapid and adaptive evolution of MHC genes under parasite selection in experimental vertebrate populations. *Nat Commun 3* (2012), 621–621.

[32] Erlich, H. HLA DNA typing: past, present, and future. *Tissue Antigens 80*, 1 (Jul 2012), 1–11.

[33] Evseeva, I., Nicodemus, K. K., Bonilla, C., Tonks, S., and Bodmer, W. F. Linkage disequilibrium and age of HLA region SNPs in relation to classic HLA gene alleles within Europe. *Eur J Hum Genet 18*, 8 (Aug 2010), 924–932.

[34] Excoffier, L., and Slatkin, M. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol Biol Evol 12*, 5 (Sep 1995), 921–927.

[35] Falush, D., Stephens, M., and Pritchard, J. K. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics 164*, 4 (Aug 2003), 1567–1587.

[36] Gabriel, C., Danzer, M., Hackl, C., Kopal, G., Hufnagl, P., Hofer, K., Polin, H., Stabentheiner, S., and Pröll, J. Rapid high-throughput human leukocyte antigen typing by massively parallel pyrosequencing for high-resolution allele identification. *Hum Immunol 70*, 11 (Nov 2009), 960–964.

[37] Gao, X., and Starmer, J. Human population structure detection via multilocus genotype clustering. *BMC Genet 8* (2007), 34–34.

[38] Gibson, G. Rare and common variants: twenty arguments. *Nat Rev Genet 13*, 2 (Feb 2011), 135–145.

[39] Gogarten, S. M., Bhangale, T., Conomos, M. P., Laurie, C. A., McHugh, C. P., Painter, I., Zheng, X., Crosslin, D. R., Levine, D., Lumley, T., Nelson, S. C., Rice, K., Shen, J., Swarnkar, R., Weir, B. S., and Laurie, C. C. GWASTools: an R/Bioconductor package for quality control and analysis of Genome-Wide Association Studies. *Bioinformatics* (Oct 2012).

[40] Hanis, C. L., Chakraborty, R., Ferrell, R. E., and Schull, W. J. Individual admixture estimates: disease associations and individual risk of diabetes and gallbladder disease among Mexican-Americans in Starr County, Texas. *Am J Phys Anthropol 70*, 4 (Aug 1986), 433–441.

[41] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2 ed. Springer, 2009.

[42] Hawley, M. E., and Kidd, K. K. HAPLO: A program using the EM algorithm to estimate the frequencies of multi-site haplotypes. *Journal of Heredity 86*, 5 (1995), 409–411.

[43] Hetherington, S., Hughes, A. R., Mosteller, M., Shortino, D., Baker, K. L., Spreen, W., Lai, E., Davies, K., Handley, A., Dow, D. J., Fling, M. E., Stocum, M., Bowman, C., Thurmond, L. M., and Roses, A. D. Genetic variations in HLA-B region and hypersensitivity reactions to abacavir. *Lancet 359*, 9312 (Mar 2002), 1121–1122.

[44] Hill, W. G. Tests for association of gene frequencies at several loci in random mating diploid populations. *Biometrics 31*, 4 (Dec 1975), 881–888.

[45] Hindorff, L. A., Sethupathy, P., Junkins, H. A., Ramos, E. M., Mehta, J. P., Collins, F. S., and Manolio, T. A. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proc Natl Acad Sci U S A 106*, 23 (Jun 2009), 9362–9367.

[46] Holcomb, C. L., Höglund, B., Anderson, M. W., Blake, L. A., Böhme, I., Egholm, M., Ferriola, D., Gabriel, C., Gelber, S. E., Goodridge, D., Hawbecker, S., Klein, R., Ladner, M., Lind, C., Monos, D., Pando, M. J., Pröll, J., Sayer, D. C., Schmitz-Agheguian, G., Simen, B. B., Thiele, B., Trachtenberg, E. A., Tyan, D. B., Wassmuth, R., White, S., and Erlich, H. A. A multi-site study using high-resolution HLA genotyping by next generation sequencing. *Tissue Antigens 77*, 3 (Mar 2011), 206–217.

[47] Horton, R., Wilming, L., Rand, V., Lovering, R. C., Bruford, E. A., Khodiyar, V. K., Lush, M. J., Povey, S., Talbot, C. C., Wright, M. W., Wain, H. M., Trowsdale, J., Ziegler, A., and Beck, S. Gene map of the extended human MHC. *Nat Rev Genet 5*, 12 (Dec 2004), 889–899.

[48] Howie, B. N., Donnelly, P., and Marchini, J. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet 5*, 6 (Jun 2009).

[49] Hung, S. I., Chung, W. H., Liou, L. B., Chu, C. C., Lin, M., Huang, H. P., Lin, Y. L., Lan, J. L., Yang, L. C., Hong, H. S., Chen, M. J., Lai, P. C., Wu, M. S., Chu, C. Y., Wang, K. H., Chen, C. H., Fann, C. S., Wu, J. Y., and Chen, Y. T. HLA-B*5801 allele as a genetic marker for severe cutaneous adverse reactions caused by allopurinol. *Proc Natl Acad Sci USA 102*, 11 (Mar 2005), 4134–4139.

[50] International HapMap 3 Consortium, Altshuler, D. M., Gibbs, R. A., Peltonen, L., Altshuler, D. M., Gibbs, R. A., Peltonen, L., Dermitzakis, E., Schaffner, S. F., Yu, F., Peltonen, L., Dermitzakis, E., Bonnen, P. E., Altshuler, D. M., Gibbs, R. A., de Bakker, P. I., Deloukas, P., Gabriel, S. B., Gwilliam, R., Hunt, S., Inouye, M., Jia, X., Palotie, A., Parkin, M., Whittaker, P., Yu, F., Chang, K., Hawes, A., Lewis, L. R., Ren, Y., Wheeler, D., Gibbs, R. A., Muzny, D. M., Barnes, C., Darvishi, K., Hurles, M., Korn, J. M., Kristiansson, K., Lee, C., McCarrol, S. A., Nemesh, J., Dermitzakis, E., Keinan, A., Montgomery, S. B., Pollack, S., Price, A. L., Soranzo, N., Bonnen, P. E., Gibbs, R. A., Gonzaga-Jauregui, C., Keinan, A., Price, A. L., Yu, F., Anttila, V., Brodeur, W., Daly, M. J., Leslie, S., McVean, G., Moutsianas, L., Nguyen, H., Schaffner, S. F., Zhang, Q., Ghori, M. J., McGinnis, R., McLaren, W., Pollack, S., Price, A. L., Schaffner, S. F., Takeuchi, F., Grossman, S. R., Shlyakhter, I., Hostetter, E. B., Sabeti, P. C., Adebamowo, C. A., Foster, M. W., Gordon, D. R., Licinio, J., Manca, M. C., Marshall, P. A., Matsuda, I., Ngare, D., Wang, V. O., Reddy, D., Rotimi, C. N., Royal, C. D., Sharp, R. R., Zeng, C., Brooks, L. D., and McEwen, J. E. Integrating common and rare genetic variation in diverse human populations. *Nature 467*, 7311 (Sep 2010), 52–58.

[51] Jacquard, A. Genetic information given by a relative. *Biometrics 28*, 4 (Dec 1972), 1101–1114.

[52] Jacquard, A. Heritability: one word, three concepts. *Biometrics 39*, 2 (Jun 1983), 465–477.

[53] Jeffreys, A. J., Ritchie, A., and Neumann, R. High resolution analysis of haplotype diversity and meiotic crossover in the human TAP2 recombination hotspot. *Hum Mol Genet 9*, 5 (Mar 2000), 725–733.

[54] Kearns, M. Thoughts on hypothesis boosting. *Unpublished manuscript (Machine Learning class project)* (December 1988).

[55] Kimura, M. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983.

[56] Lango Allen, H., Estrada, K., Lettre, G., Berndt, S. I., Weedon, M. N., Rivadeneira, F., Willer, C. J., Jackson, A. U., Vedantam, S., Raychaudhuri, S., Ferreira, T., Wood, A. R., Weyant, R. J., Segrè, A. V., Speliotes, E. K., Wheeler, E., Soranzo, N., Park, J. H., Yang, J., Gudbjartsson, D., Heard-Costa, N. L., Randall, J. C., Qi, L., Vernon Smith, A., Mägi, R., Pastinen, T., Liang, L., Heid, I. M., Luan, J., Thorleifsson, G., Winkler, T. W., Goddard, M. E., Sin Lo, K., Palmer, C., Workalemahu, T., Aulchenko, Y. S., Johansson, A., Zillikens, M. C., Feitosa, M. F., Esko, T., Johnson, T., Ketkar, S., Kraft, P., Mangino, M., Prokopenko, I., Absher, D., Albrecht, E., Ernst, F., Glazer, N. L., Hayward, C., Hottenga, J. J., Jacobs, K. B., Knowles, J. W., Kutalik, Z., Monda, K. L., Polasek, O., Preuss, M., Rayner, N. W., Robertson, N. R., Steinthorsdottir, V., Tyrer, J. P., Voight, B. F., Wiklund, F., Xu, J., Zhao, J. H., Nyholt, D. R., Pellikka, N., Perola, M., Perry, J. R., Surakka, I., Tammesoo, M. L., Altmaier, E. L., Amin, N., Aspelund, T., Bhangale, T., Boucher, G., Chasman, D. I., Chen, C., Coin, L., Cooper, M. N., Dixon, A. L., Gibson, Q., Grundberg, E., Hao, K., Juhani Junttila, M., Kaplan, L. M., Kettunen, J., König, I. R., Kwan, T., Lawrence, R. W., Levinson, D. F., Lorentzon, M., McKnight, B., Morris, A. P., Müller, M., Suh Ngwa, J., Purcell, S., Rafelt, S., Salem, R. M., Salvi, E., Sanna, S., Shi, J., Sovio, U., Thompson, J. R., Turchin, M. C., Vandenput, L., Verlaan, D. J., Vitart, V., White, C. C., Ziegler, A., Almgren, P., Balmforth, A. J., Campbell, H., Citterio, L., De Grandi, A., Dominiczak, A., Duan, J., Elliott, P., Elosua, R., Eriksson, J. G., Freimer, N. B., Geus, E. J., Glorioso, N., Haiqing, S., Hartikainen, A. L., Havulinna, A. S., Hicks, A. A., Hui, J., Igl, W., Illig, T., Jula, A., Kajantie, E., Kilpeläinen, T. O., Koiranen, M., Kolcic, I., Koskinen, S., Kovacs, P., Laitinen, J., Liu, J., Lokki, M. L., Marusic, A., Maschio, A., Meitinger, T., Mulas, A., Paré, G., Parker, A. N., Peden, J. F., Petersmann, A., Pichler, I., Pietiläinen, K. H., Pouta, A., Ridderstråle, M., Rotter, J. I., Sambrook, J. G., Sanders, A. R., Schmidt, C. O., Sinisalo, J., Smit, J. H., Stringham, H. M., Bragi Walters, G., Widen, E., Wild, S. H., Willemsen, G., Zagato, L., Zgaga, L., Zitting, P., Alavere, H., Farrall, M., McArdle, W. L., Nelis, M., Peters, M. J., Ripatti, S., van Meurs, J. B., Aben, K. K., Ardlie, K. G., Beckmann, J. S., Beilby, J. P., Bergman, R. N., Bergmann, S., Collins,

F. S., Cusi, D., den Heijer, M., Eiriksdottir, G., Gejman, P. V., Hall, A. S., Hamsten, A., Huikuri, H. V., Iribarren, C., Kähönen, M., Kaprio, J., Kathiresan, S., Kiemeney, L., Kocher, T., Launer, L. J., Lehtimäki, T., Melander, O., Mosley, T. H., Musk, A. W., Nieminen, M. S., O'Donnell, C. J., Ohlsson, C., Oostra, B., Palmer, L. J., Raitakari, O., Ridker, P. M., Rioux, J. D., Rissanen, A., Rivolta, C., Schunkert, H., Shuldiner, A. R., Siscovick, D. S., Stumvoll, M., Tönjes, A., Tuomilehto, J., van Ommen, G. J., Viikari, J., Heath, A. C., Martin, N. G., Montgomery, G. W., Province, M. A., Kayser, M., Arnold, A. M., Atwood, L. D., Boerwinkle, E., Chanock, S. J., Deloukas, P., Gieger, C., Grönberg, H., Hall, P., Hattersley, A. T., Hengstenberg, C., Hoffman, W., Lathrop, G. M., Salomaa, V., Schreiber, S., Uda, M., Waterworth, D., Wright, A. F., Assimes, T. L., Barroso, I., Hofman, A., Mohlke, K. L., Boomsma, D. I., Caulfield, M. J., Cupples, L. A., Erdmann, J., Fox, C. S., Gudnason, V., Gyllensten, U., Harris, T. B., Hayes, R. B., Jarvelin, M. R., Mooser, V., Munroe, P. B., Ouwehand, W. H., Penninx, B. W., Pramstaller, P. P., Quertermous, T., Rudan, I., Samani, N. J., Spector, T. D., Völzke, H., Watkins, H., Wilson, J. F., Groop, L. C., Haritunians, T., Hu, F. B., Kaplan, R. C., Metspalu, A., North, K. E., Schlessinger, D., Wareham, N. J., Hunter, D. J., O'Connell, J. R., Strachan, D. P., Wichmann, H. E., Borecki, I. B., van Duijn, C. M., Schadt, E. E., Thorsteinsdottir, U., Peltonen, L., Uitterlinden, A. G., Visscher, P. M., Chatterjee, N., Loos, R. J., Boehnke, M., McCarthy, M. I., Ingelsson, E., Lindgren, C. M., Abecasis, G. R., Stefansson, K., Frayling, T. M., and Hirschhorn, J. N. Hundreds of variants clustered in genomic loci and biological pathways affect human height. *Nature 467*, 7317 (Oct 2010), 832–838.

[57] Laurie, C. C., Doheny, K. F., Mirel, D. B., Pugh, E. W., Bierut, L. J., Bhangale, T., Boehm, F., Caporaso, N. E., Cornelis, M. C., Edenberg, H. J., Gabriel, S. B., Harris, E. L., Hu, F. B., Jacobs, K. B., Kraft, P., Landi, M. T., Lumley, T., Manolio, T. A., McHugh, C., Painter, I., Paschall, J., Rice, J. P., Rice, K. M., Zheng, X., Weir, B. S., and GENEVA Investigators. Quality control and quality assurance in genotypic data for genome-wide association studies. *Genet Epidemiol 34*, 6 (Sep 2010), 591–602.

[58] Lee, C., Abdool, A., and Huang, C. H. PCA-based population structure inference with generic clustering algorithms. *BMC Bioinformatics 10 Suppl 1* (2009).

[59] Leslie, S., Donnelly, P., and McVean, G. A statistical method for predicting classical HLA alleles from SNP data. *Am J Hum Genet 82*, 1 (Jan 2008), 48–56.

[60] Levin, A. M., Adrianto, I., Datta, I., Iannuzzi, M. C., Trudeau, S., McKeigue,

P., Montgomery, C. G., and Rybicki, B. A. Evaluation of classical HLA allele prediction methods in a sample of African Americans and European Americans. *Submitted* (2013).

[61] Li, N., and Stephens, M. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics 165*, 4 (Dec 2003), 2213–2233.

[62] Li, S. S., Cheng, J. J., and Zhao, L. P. Empirical vs bayesian approach for estimating haplotypes from genotypes of unrelated individuals. *BMC Genet 8* (2007), 2–2.

[63] Li, S. S., Khalid, N., Carlson, C., and Zhao, L. P. Estimating haplotype frequencies and standard errors for multiple single nucleotide polymorphisms. *Biostatistics 4*, 4 (Oct 2003), 513–522.

[64] Li, S. S., Wang, H., Smith, A., Zhang, B., Zhang, X. C., Schoch, G., Geraghty, D., Hansen, J. A., and Zhao, L. P. Predicting multiallelic genes using unphased and flanking single nucleotide polymorphisms. *Genet Epidemiol 35*, 2 (Feb 2011), 85–92.

[65] Li, Y., Willer, C. J., Ding, J., Scheet, P., and Abecasis, G. R. MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet Epidemiol 34*, 8 (Dec 2010), 816–834.

[66] Long, J. C., Williams, R. C., and Urbanek, M. An E-M algorithm and testing strategy for multiple-locus haplotypes. *Am J Hum Genet 56*, 3 (Mar 1995), 799–810.

[67] Maher, B. Personal genomes: The case of the missing heritability. *Nature 456*, 7218 (Nov 2008), 18–21.

[68] Maiers, M., Gragert, L., and Klitz, W. High-resolution HLA alleles and haplotypes in the United States population. *Human Immunology 68*, 9 (2007), 779 – 788.

[69] Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., McCarthy, M. I., Ramos, E. M., Cardon, L. R., Chakravarti, A., Cho, J. H., Guttmacher, A. E., Kong, A., Kruglyak, L., Mardis, E., Rotimi, C. N., Slatkin, M., Valle, D., Whittemore, A. S., Boehnke, M., Clark, A. G., Eichler, E. E., Gibson, G., Haines, J. L., Mackay, T. F., McCarroll, S. A., and Visscher, P. M. Finding the missing heritability of complex diseases. *Nature 461*, 7265 (Oct 2009), 747–753.

[70] Marchini, J., and Howie, B. Genotype imputation for genome-wide association studies. *Nat Rev Genet 11*, 7 (2010), 499–511.

[71] McClellan, J., and King, M.-C. Genetic heterogeneity in human disease. *Cell 141*, 2 (2010), 210 – 217.

[72] McVean, G. A genealogical interpretation of principal components analysis. *PLoS Genet 5*, 10 (Oct 2009).

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

[74] McVean, G. A., Myers, S. R., Hunt, S., Deloukas, P., Bentley, D. R., and Donnelly, P. The fine-scale structure of recombination rate variation in the human genome. *Science 304*, 5670 (Apr 2004), 581–584.

[75] Media, O. *Hadoop: The Definitive Guide*. ISBN 978-1-4493-3877-0. White, Tom, 2012.

[76] Menozzi, P., Piazza, A., and Cavalli-Sforza, L. Synthetic maps of human gene frequencies in Europeans. *Science 201*, 4358 (Sep 1978), 786–792.

[77] Mungall, A. J., Palmer, S. A., Sims, S. K., Edwards, C. A., Ashurst, J. L., Wilming, L., Jones, M. C., Horton, R., Hunt, S. E., Scott, C. E., Gilbert, J. G., Clamp, M. E., Bethel, G., Milne, S., Ainscough, R., Almeida, J. P., Ambrose, K. D., Andrews, T. D., Ashwell, R. I., Babbage, A. K., Bagguley, C. L., Bailey, J., Banerjee, R., Barker, D. J., Barlow, K. F., Bates, K., Beare, D. M., Beasley, H., Beasley, O., Bird, C. P., Blakey, S., Bray-Allen, S., Brook, J., Brown, A. J., Brown, J. Y., Burford, D. C., Burrill, W., Burton, J., Carder, C., Carter, N. P., Chapman, J. C., Clark, S. Y., Clark, G., Clee, C. M., Clegg, S., Cobley, V., Collier, R. E., Collins, J. E., Colman, L. K., Corby, N. R., Coville, G. J., Culley, K. M., Dhami, P., Davies, J., Dunn, M., Earthrowl, M. E., Ellington, A. E., Evans, K. A., Faulkner, L., Francis, M. D., Frankish, A., Frankland, J., French, L., Garner, P., Garnett, J., Ghori, M. J., Gilby, L. M., Gillson, C. J., Glithero, R. J., Grafham, D. V., Grant, M., Gribble, S., Griffiths, C., Griffiths, M., Hall, R., Halls, K. S., Hammond, S., Harley, J. L., Hart, E. A., Heath, P. D., Heathcott, R., Holmes, S. J., Howden, P. J., Howe, K. L., Howell, G. R., Huckle, E., Humphray, S. J., Humphries, M. D., Hunt, A. R., Johnson, C. M., Joy, A. A., Kay, M., Keenan, S. J., Kimberley, A. M., King, A., Laird, G. K., Langford, C., Lawlor, S., Leongamornlert, D. A., Leversha, M., Lloyd, C. R., Lloyd, D. M., Loveland, J. E., Lovell, J., Martin, S., Mashreghi-Mohammadi, M., Maslen, G. L., Matthews, L., McCann, O. T., McLaren, S. J., McLay,

K., McMurray, A., Moore, M. J., Mullikin, J. C., Niblett, D., Nickerson, T., Novik, K. L., Oliver, K., Overton-Larty, E. K., Parker, A., Patel, R., Pearce, A. V., Peck, A. I., Phillimore, B., Phillips, S., Plumb, R. W., Porter, K. M., Ramsey, Y., Ranby, S. A., Rice, C. M., Ross, M. T., Searle, S. M., Sehra, H. K., Sheridan, E., Skuce, C. D., Smith, S., Smith, M., Spraggon, L., Squares, S. L., Steward, C. A., Sycamore, N., Tamlyn-Hall, G., Tester, J., Theaker, A. J., Thomas, D. W., Thorpe, A., Tracey, A., Tromans, A., Tubby, B., Wall, M., Wallis, J. M., West, A. P., White, S. S., Whitehead, S. L., Whittaker, H., Wild, A., Willey, D. J., Wilmer, T. E., Wood, J. M., Wray, P. W., Wyatt, J. C., Young, L., Younger, R. M., Bentley, D. R., Coulson, A., Durbin, R., Hubbard, T., Sulston, J. E., Dunham, I., Rogers, J., and Beck, S. The DNA sequence and analysis of human chromosome 6. *Nature 425*, 6960 (Oct 2003), 805–811.

[78] Myers, S., Bottolo, L., Freeman, C., McVean, G., and Donnelly, P. A fine-scale map of recombination rates and hotspots across the human genome. *Science 310*, 5746 (Oct 2005), 321–324.

[79] Nejentsev, S., Howson, J. M., Walker, N. M., Szeszko, J., Field, S. F., Stevens, H. E., Reynolds, P., Hardy, M., King, E., Masters, J., Hulme, J., Maier, L. M., Smyth, D., Bailey, R., Cooper, J. D., Ribas, G., Campbell, R. D., Clayton, D. G., Todd, J. A., and Wellcome Trust Case Control Consortium. Localization of type 1 diabetes susceptibility to the MHC class I genes HLA-B and HLA-A. *Nature 450*, 7171 (Dec 2007), 887–892.

[80] Niu, T., Qin, Z. S., Xu, X., and Liu, J. S. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am J Hum Genet 70*, 1 (Jan 2002), 157–169.

[81] Novembre, J., Johnson, T., Bryc, K., Kutalik, Z., Boyko, A. R., Auton, A., Indap, A., King, K. S., Bergmann, S., Nelson, M. R., Stephens, M., and Bustamante, C. D. Genes mirror geography within Europe. *Nature 456*, 7218 (Nov 2008), 98–101.

[82] Novembre, J., and Stephens, M. Interpreting principal component analyses of spatial population genetic variation. *Nat Genet 40*, 5 (May 2008), 646–649.

[83] Patterson, N., Price, A. L., and Reich, D. Population structure and eigenanalysis. *PLoS Genet 2*, 12 (Dec 2006).

[84] Pemberton, T. J., Wang, C., Li, J. Z., and Rosenberg, N. A. Inference of unexpected genetic relatedness among individuals in HapMap Phase III. *Am J Hum Genet 87*, 4 (Oct 2010), 457–464.

[85] Price, A. L., Patterson, N., Yu, F., Cox, D. R., Waliszewska, A., McDonald, G. J., Tandon, A., Schirmer, C., Neubauer, J., Bedoya, G., Duque, C., Villegas, A., Bortolini, M. C., Salzano, F. M., Gallo, C., Mazzotti, G., Tello-Ruiz, M., Riba, L., Aguilar-Salinas, C. A., Canizales-Quinteros, S., Menjivar, M., Klitz, W., Henderson, B., Haiman, C. A., Winkler, C., Tusie-Luna, T., Ruiz-Linares, A., and Reich, D. A genome-wide admixture map for Latino populations. *Am J Hum Genet 80*, 6 (Jun 2007), 1024–1036.

[86] Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. Principal components analysis corrects for stratification in genome-wide association studies. *Nat Genet 38*, 8 (Aug 2006), 904–909.

[87] Price, A. L., Tandon, A., Patterson, N., Barnes, K. C., Rafaels, N., Ruczinski, I., Beaty, T. H., Mathias, R., Reich, D., and Myers, S. Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. *PLoS Genet 5*, 6 (Jun 2009).

[88] Price, A. L., Zaitlen, N. A., Reich, D., and Patterson, N. New approaches to population stratification in genome-wide association studies. *Nat Rev Genet 11*, 7 (Jul 2010), 459–463.

[89] Pritchard, J. K., and Cox, N. J. The allelic architecture of human disease genes: common disease-common variant...or not? *Hum Mol Genet 11*, 20 (Oct 2002), 2417–2423.

[90] Pritchard, J. K., Stephens, M., and Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics 155*, 2 (Jun 2000), 945–959.

[91] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., de Bakker, P. I., Daly, M. J., and Sham, P. C. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am J Hum Genet 81*, 3 (Sep 2007), 559–575.

[92] Qin, Z. S., Niu, T., and Liu, J. S. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *Am J Hum Genet 71*, 5 (Nov 2002), 1242–1247.

[93] Raychaudhuri, S., Sandor, C., Stahl, E. A., Freudenberg, J., Lee, H. S., Jia, X., Alfredsson, L., Padyukov, L., Klareskog, L., Worthington, J., Siminovitch, K. A., Bae, S. C., Plenge, R. M., Gregersen, P. K., and de Bakker, P. I. Five amino acids in three HLA proteins explain most of the association between MHC and seropositive rheumatoid arthritis. *Nat Genet 44*, 3 (2012), 291–296.

[94] Robinson, J., Mistry, K., McWilliam, H., Lopez, R., Parham, P., and Marsh, S. The IMGT/HLA database. *Nucleic Acids Research* (2010).

[95] Scally, A., and Durbin, R. Revising the human mutation rate: implications for understanding human evolution. *Nat Rev Genet 13*, 10 (Oct 2012), 745–753.

[96] Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., and Nolan, G. P. Computational solutions to large-scale data management and analysis. *Nat Rev Genet 11*, 9 (Sep 2010), 647–657.

[97] Scheet, P., and Stephens, M. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am J Hum Genet 78*, 4 (Apr 2006), 629–644.

[98] Setty, M. N., Gusev, A., and Pe'er, I. HLA type inference via haplotypes identical by descent. *J Comput Biol 18*, 3 (Mar 2011), 483–493.

[99] Shiina, T., Hosomichi, K., Inoko, H., and Kulski, J. The HLA genomic loci map: expression, interaction, diversity and disease. *Journal of human genetics 54*, 1 (2009), 15–39.

[100] Slatkin, M. Inbreeding coefficients and coalescence times. *Genet Res 58*, 2 (Oct 1991), 167–175.

[101] Slatkin, M. Epigenetic inheritance and the missing heritability problem. *Genetics 182*, 3 (Jul 2009), 845–850.

[102] Stephens, M., and Scheet, P. Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *Am J Hum Genet 76*, 3 (Mar 2005), 449–462.

[103] Tang, H., Peng, J., Wang, P., and Risch, N. J. Estimation of individual admixture: analytical and study design considerations. *Genet Epidemiol 28*, 4 (May 2005), 289–301.

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

[105] Thompson, E. A. The estimation of pairwise relationships. *Ann Hum Genet 39*, 2 (Oct 1975), 173–188.

[106] Thornton, T., Tang, H., Hoffmann, T. J., Ochs-Balcom, H. M., Caan, B. J., and Risch, N. Estimating kinship in admixed populations. *Am J Hum Genet 91*, 1 (Jul 2012), 122–138.

[107] Vandiedonck, C., and Knight, J. C. The human major histocompatibility complex as a paradigm in genomics research. *Brief Funct Genomic Proteomic 8*, 5 (Sep 2009), 379–394.

[108] Visscher, P. M., Brown, M. A., McCarthy, M. I., and Yang, J. Five years of GWAS discovery. *Am J Hum Genet 90*, 1 (Jan 2012), 7–24.

[109] Visscher, P. M., Hill, W. G., and Wray, N. R. Heritability in the genomics era–concepts and misconceptions. *Nat Rev Genet 9*, 4 (Apr 2008), 255–266.

[110] Wakeley, J. The limits of theoretical population genetics. *Genetics 169*, 1 (2005), 1–7.

[111] Wang, J. Unbiased relatedness estimation in structured populations. *Genetics 187*, 3 (Mar 2011), 887–901.

[112] Weir, B. S., and Cockerham, C. C. Estimating F-statistics for the analysis of population structure. *Evolution 38*, 6 (1984), pp. 1358–1370.

[113] Weir, B. S., and Hill, W. G. Estimating F-statistics. *Annu Rev Genet 36* (2002), 721–750.

[114] Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature 447*, 7145 (Jun 2007), 661–678.

[115] Williams, A., Patterson, N., Glessner, J., Hakonarson, H., and Reich, D. Phasing of many thousands of genotyped samples. *American journal of human genetics 91*, 2 (2012), 238–251.

[116] Wright, S. Isolation by distance. *Genetics 2*, 28 (1943), 114–38.

[117] Xie, M., Li, J., and Jiang, T. Accurate HLA type inference using a weighted similarity graph. *BMC Bioinformatics 11 Suppl 11* (2010).

[118] Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., Madden, P. A., Heath, A. C., Martin, N. G., Montgomery, G. W., Goddard, M. E., and Visscher, P. M. Common SNPs explain a large proportion of the heritability for human height. *Nat Genet 42*, 7 (Jul 2010), 565–569.

[119] Yang, J., Manolio, T. A., Pasquale, L. R., Boerwinkle, E., Caporaso, N., Cunningham, J. M., de Andrade, M., Feenstra, B., Feingold, E., Hayes, M. G., Hill, W. G., Landi, M. T., Alonso, A., Lettre, G., Lin, P., Ling, H., Lowe, W., Mathias, R. A., Melbye, M., Pugh, E., Cornelis, M. C., Weir, B. S., Goddard, M. E., and Visscher, P. M. Genome partitioning of genetic variation for complex traits using common snps. *Nat Genet 43*, 6 (Jun 2011), 519–525.

[120] Zheng, X., Levine, D., Shen, J., Gogarten, S. M., Laurie, C., and Weir, B. S. A high-performance computing toolset for relatedness and principal component analysis of SNP data. *Bioinformatics* (Oct 2012).

[121] Zheng, X., Shen, J., Cox, C., Wakefield, J. C., Ehm, M. G., Nelson, M. R., and Weir, B. S. HIBAG – HLA genotype imputation with attribute bagging. *Pharmacogenomics J* (May 2013).

[122] Zhu, X., Li, S., Cooper, R. S., and Elston, R. C. A unified association analysis approach for family and unrelated samples correcting for stratification. *Am J Hum Genet 82*, 2 (Feb 2008), 352–365.

Appendix A

Supplementary Information for HLA Imputation

A.1 Tables

Table S1.1: The SNP list used by HLA*IMP when Illumina 1M platform is specified.

Locus (# of SNPs)	Marker list
HLA–A (50)	rs1737083, rs9391630, rs4947236, rs9258275, rs1737060, rs1633041, rs1737043, rs1633021, rs9258437, rs2517922, rs1632988, rs2523409, rs1610663, rs1610707, rs3115630, rs915669, rs2517892, rs2734999, rs3115629, rs2394185, rs9380146, rs9258631, rs2734979, rs2508049, rs2508046, rs3094159, rs2734959, rs2517817, rs2517904, rs2517891, rs1611493, rs2517755, rs2860580, rs2517722, rs7745413, rs7747253, rs7739434, rs9260759, rs5009448, rs2735076, rs2735071, rs3132685, rs5025708, rs3121597, rs166326, rs16896944, rs3132129, rs1150738, rs1245371, rs11965797
HLA–B (39)	rs1265156, rs4713438, rs3130466, rs9295967, rs9263979, rs6904669, rs2524099, rs2524123, rs9461688, rs2524229, rs9295976, rs9265797, rs2442719, rs2596501, rs1058026, rs2523591, rs2523590, rs2523589, rs2523578, rs2523557, rs2844573, rs6936035, rs3094600, rs9266689, rs2442752, rs2596560, rs2523471, rs2256175, rs9266845, rs3094738, rs2596460, rs3094228, rs2395030, rs2284178, rs2523674, rs2905722, rs2523710, rs2534671, rs2855807
HLA–C (27)	rs9263719, rs3823417, rs1265099, rs2074478, rs1265094, rs130075, rs9263800, rs1265158, rs3130467, rs3130531, rs2844623, rs9264532, rs2524099, rs2395471, rs2249742, rs2524084, rs12111032, rs6906846, rs6917363, rs3915971, rs7761965, rs9264942, rs7453967, rs2442719, rs2523535, rs2844529, rs3763288

HLA–DRB1 (50)	rs6907322, rs2273017, rs2050190, rs2076536, rs2073045, rs2050189, rs4248166, rs3817964, rs3793126, rs10947262, rs3806155, rs6932542, rs2395163, rs3135363, rs2027856, rs3129882, rs2239804, rs6919855, rs9268862, rs5020946, rs9270623, rs4599680, rs615672, rs2858867, rs482044, rs660895, rs2097431, rs9273012, rs6906021, rs3134975, rs9275184, rs7774434, rs2647044, rs9275424, rs9275425, rs9275572, rs7745656, rs2858332, rs3957148, rs3873444, rs9275602, rs3892710, rs5024431, rs12177980, rs7755596, rs17500468, rs10807113, rs4947347, rs2071550, rs6903130
HLA–DQB1 (34)	rs3117099, rs3817964, rs3763305, rs743862, rs3129867, rs2239802, rs4599680, rs482044, rs13207945, rs9272723, rs6927022, rs6906021, rs1063355, rs2300825, rs3891175, rs3828796, rs7755224, rs3134975, rs2856691, rs2856683, rs7774434, rs9275224, rs9275313, rs3135006, rs9275555, rs3104402, rs3104405, rs9275601, rs3892710, rs6935940, rs2395246, rs17500510, rs2071800, rs719654

Table S1.2: The sensitivity (SEN), specificity (SPE), positive predictive value (PPV) and negative predictive value (NPV) calculated from validation samples for each four-digit HLA allele with call threshold 0.5, when study data of European ancestry were divided to training and validation parts with approximately equal sizes. The SNP markers in the intersect of Illumina platforms were used.

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
HLA–A: Overall accuracy: 98.7%											
01:01	270	0.1429	271	0.1486	99.6	100.0	100.0	100.0	100.0	100.0	–
02:01	503	0.2661	533	0.2922	99.2	99.7	100.0	99.5	98.9	100.0	–
02:02	3	0.0016	3	0.0016	100.0	99.9	66.7	100.0	100.0	99.9	02:05 (100)
02:05	19	0.0101	19	0.0104	100.0	99.9	100.0	99.9	95.0	100.0	–
03:01	252	0.1333	234	0.1283	99.6	99.8	100.0	99.8	98.7	100.0	–
11:01	127	0.0672	120	0.0658	99.2	100.0	100.0	100.0	100.0	100.0	–
23:01	39	0.0206	42	0.0230	100.0	100.0	100.0	100.0	100.0	100.0	–
24:02	170	0.0899	141	0.0773	97.9	99.7	100.0	99.7	96.5	100.0	–
25:01	55	0.0291	50	0.0274	98.0	99.6	91.8	99.8	91.8	99.8	26:01 (100)
26:01	74	0.0392	70	0.0384	98.6	99.4	92.8	99.7	92.8	99.7	25:01 (80)
29:01	10	0.0053	10	0.0055	80.0	100.0	100.0	100.0	100.0	100.0	–
29:02	63	0.0333	57	0.0312	98.2	100.0	100.0	100.0	100.0	100.0	–
30:01	30	0.0159	27	0.0148	100.0	100.0	100.0	100.0	100.0	100.0	–
30:02	19	0.0101	17	0.0093	100.0	100.0	100.0	100.0	100.0	100.0	–
31:01	46	0.0243	45	0.0247	93.3	100.0	100.0	100.0	100.0	100.0	–
32:01	69	0.0365	66	0.0362	100.0	100.0	100.0	100.0	100.0	100.0	–
33:01	13	0.0069	12	0.0066	100.0	100.0	100.0	100.0	100.0	100.0	–
33:03	9	0.0048	9	0.0049	100.0	100.0	100.0	100.0	100.0	100.0	–
34:02	2	0.0011	1	0.0005	100.0	100.0	100.0	100.0	100.0	100.0	–
66:01	12	0.0063	11	0.0060	100.0	99.9	90.9	100.0	100.0	99.9	26:01 (100)
68:01	54	0.0286	58	0.0318	96.6	100.0	100.0	100.0	100.0	100.0	–
68:02	13	0.0069	11	0.0060	100.0	100.0	100.0	100.0	100.0	100.0	–
69:01	2	0.0011	2	0.0011	50.0	100.0	100.0	100.0	100.0	100.0	–
HLA–B: Overall accuracy: 97.8%											
07:02	319	0.1214	291	0.1157	97.3	99.9	100.0	99.9	99.3	100.0	–
07:05	8	0.0030	7	0.0028	100.0	100.0	100.0	100.0	100.0	100.0	–
08:01	239	0.0909	262	0.1041	97.3	100.0	100.0	100.0	100.0	100.0	–
13:02	87	0.0331	85	0.0338	94.1	100.0	100.0	100.0	100.0	100.0	–
14:01	17	0.0065	17	0.0068	82.4	100.0	100.0	100.0	100.0	100.0	–
14:02	61	0.0232	60	0.0238	96.7	100.0	100.0	100.0	100.0	100.0	–
15:01	147	0.0559	160	0.0636	98.1	99.8	100.0	99.8	97.5	100.0	–
15:17	10	0.0038	10	0.0040	100.0	100.0	100.0	100.0	100.0	100.0	–
15:18	5	0.0019	4	0.0016	100.0	100.0	75.0	100.0	100.0	100.0	15:01 (100)
18:01	140	0.0533	152	0.0604	93.4	99.6	99.3	99.6	94.6	100.0	38:01 (100)
27:02	24	0.0091	23	0.0091	95.7	99.7	72.7	100.0	100.0	99.8	27:05 (83)
27:05	98	0.0373	95	0.0378	96.8	99.8	100.0	99.8	94.8	100.0	–
35:01	149	0.0567	125	0.0497	88.0	99.4	95.5	99.6	91.3	99.8	35:02 (60)
35:02	24	0.0091	24	0.0095	83.3	99.9	100.0	99.9	87.0	100.0	–
35:03	54	0.0205	61	0.0242	91.8	99.6	82.1	100.0	100.0	99.6	35:01 (80)
35:08	19	0.0072	18	0.0072	55.6	99.9	90.0	99.9	81.8	100.0	35:01 (100)
37:01	28	0.0107	26	0.0103	100.0	100.0	100.0	100.0	100.0	100.0	–
38:01	70	0.0266	74	0.0294	82.4	100.0	100.0	100.0	98.4	100.0	–
39:01	38	0.0145	35	0.0139	80.0	99.9	89.3	100.0	100.0	99.9	18:01 (67)
39:06	12	0.0046	11	0.0044	63.6	100.0	100.0	100.0	100.0	100.0	–
39:24	2	0.0008	1	0.0004	100.0	100.0	100.0	100.0	100.0	100.0	–
40:01	105	0.0400	98	0.0390	98.0	100.0	100.0	100.0	100.0	100.0	–
40:02	42	0.0160	47	0.0187	87.2	100.0	100.0	100.0	100.0	100.0	–
40:06	3	0.0011	3	0.0012	66.7	100.0	100.0	100.0	100.0	100.0	–
41:01	12	0.0046	11	0.0044	90.9	100.0	100.0	100.0	100.0	100.0	–
41:02	17	0.0065	15	0.0060	93.3	100.0	100.0	100.0	100.0	100.0	–

Continued on next page ...

Table S1.2 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
44:02	226	0.0860	212	0.0843	99.1	99.9	99.0	100.0	99.5	99.9	44:03 (50)
44:03	126	0.0479	126	0.0501	96.8	99.9	100.0	99.9	98.4	100.0	–
44:05	16	0.0061	13	0.0052	84.6	99.9	100.0	99.9	84.6	100.0	–
45:01	13	0.0049	12	0.0048	83.3	100.0	100.0	100.0	100.0	100.0	–
47:01	7	0.0027	7	0.0028	100.0	100.0	100.0	100.0	100.0	100.0	–
48:01	3	0.0011	2	0.0008	100.0	100.0	100.0	100.0	100.0	100.0	–
49:01	44	0.0167	44	0.0175	95.5	99.9	95.2	100.0	100.0	99.9	44:03 (50)
50:01	37	0.0141	35	0.0139	91.4	100.0	100.0	100.0	97.0	100.0	–
51:01	154	0.0586	122	0.0485	97.5	99.7	99.2	99.7	95.2	100.0	44:05 (100)
52:01	34	0.0129	31	0.0123	87.1	100.0	100.0	100.0	100.0	100.0	–
53:01	7	0.0027	6	0.0024	100.0	100.0	83.3	100.0	100.0	100.0	35:01 (100)
55:01	42	0.0160	44	0.0175	95.5	99.8	92.9	100.0	97.5	99.9	56:01 (100)
56:01	25	0.0095	21	0.0083	95.2	99.8	95.0	99.8	82.6	100.0	55:01 (100)
57:01	76	0.0289	77	0.0306	97.4	100.0	100.0	100.0	100.0	100.0	–
58:01	25	0.0095	22	0.0087	95.5	100.0	100.0	100.0	100.0	100.0	–
73:01	3	0.0011	2	0.0008	100.0	100.0	100.0	100.0	100.0	100.0	–
HLA–C: Overall accuracy: 99.2%											
01:02	85	0.0450	72	0.0390	98.6	100.0	100.0	100.0	100.0	100.0	–
02:02	115	0.0609	93	0.0504	100.0	100.0	100.0	100.0	100.0	100.0	–
03:02	4	0.0021	4	0.0022	100.0	100.0	100.0	100.0	100.0	100.0	–
03:03	90	0.0477	92	0.0499	91.3	99.7	97.6	99.8	95.3	99.9	03:04 (100)
03:04	113	0.0599	106	0.0575	97.2	99.7	96.1	99.9	98.0	99.8	03:03 (100)
04:01	221	0.1171	206	0.1117	99.5	100.0	100.0	100.0	100.0	100.0	–
05:01	143	0.0757	157	0.0851	99.4	99.9	99.4	99.9	99.4	99.9	08:02 (100)
06:02	176	0.0932	174	0.0944	99.4	99.9	100.0	99.9	99.4	100.0	–
07:01	278	0.1472	273	0.1480	97.8	100.0	100.0	100.0	100.0	100.0	–
07:02	228	0.1208	257	0.1394	99.2	100.0	100.0	100.0	100.0	100.0	–
07:04	38	0.0201	37	0.0201	100.0	100.0	100.0	100.0	100.0	100.0	–
08:02	48	0.0254	49	0.0266	98.0	99.9	97.9	99.9	97.9	99.9	05:01 (100)
08:03	2	0.0011	1	0.0005	100.0	100.0	100.0	100.0	100.0	100.0	–
12:02	26	0.0138	24	0.0130	95.8	100.0	100.0	100.0	100.0	100.0	–
12:03	134	0.0710	128	0.0694	99.2	99.9	99.2	100.0	100.0	99.9	06:02 (100)
14:02	21	0.0111	21	0.0114	100.0	100.0	100.0	100.0	100.0	100.0	–
15:02	43	0.0228	45	0.0244	93.3	99.9	100.0	99.9	95.5	100.0	–
15:05	8	0.0042	8	0.0043	100.0	100.0	100.0	100.0	100.0	100.0	–
16:01	63	0.0334	62	0.0336	96.8	99.9	100.0	99.9	98.4	100.0	–
16:02	6	0.0032	5	0.0027	100.0	100.0	100.0	100.0	100.0	100.0	–
16:04	5	0.0026	4	0.0022	75.0	99.9	66.7	100.0	100.0	99.9	16:01 (100)
17:01	22	0.0117	18	0.0098	94.4	99.9	100.0	99.9	89.5	100.0	–
HLA–DRB1: Overall accuracy: 94.9%											
01:01	198	0.0802	208	0.0865	97.6	99.4	99.5	99.4	94.4	100.0	01:02 (100)
01:02	33	0.0134	35	0.0146	97.1	100.0	100.0	100.0	97.1	100.0	–
01:03	18	0.0073	18	0.0075	83.3	99.4	20.0	100.0	100.0	99.5	01:01 (100)
03:01	277	0.1122	277	0.1152	94.9	99.9	99.6	99.9	99.6	100.0	15:01 (100)
04:01	179	0.0725	198	0.0824	91.9	99.2	97.3	99.3	93.2	99.8	04:07 (80)
04:02	29	0.0118	28	0.0116	82.1	99.9	100.0	99.9	92.0	100.0	–
04:03	25	0.0101	31	0.0129	64.5	99.2	15.0	100.0	100.0	99.3	04:04 (65)
04:04	74	0.0300	64	0.0266	85.9	99.2	94.5	99.3	78.2	99.9	04:01 (100)
04:05	17	0.0069	19	0.0079	57.9	100.0	90.9	100.0	100.0	100.0	04:01 (100)
04:07	25	0.0101	28	0.0116	71.4	99.5	70.0	99.8	77.8	99.7	04:01 (67)
07:01	343	0.1390	300	0.1248	94.7	99.9	99.6	99.9	99.6	100.0	04:01 (100)
08:01	65	0.0263	56	0.0233	98.2	100.0	100.0	100.0	99.1	100.0	–
08:03	4	0.0016	3	0.0012	66.7	100.0	100.0	100.0	100.0	100.0	–
08:04	5	0.0020	4	0.0017	50.0	100.0	100.0	100.0	100.0	100.0	–
09:01	26	0.0105	22	0.0092	90.9	100.0	100.0	100.0	100.0	100.0	–
10:01	16	0.0065	17	0.0071	82.4	100.0	100.0	100.0	100.0	100.0	–
11:01	165	0.0669	177	0.0736	75.7	97.7	99.3	97.6	73.5	100.0	12:01 (100)
11:02	6	0.0024	6	0.0025	50.0	100.0	66.7	100.0	100.0	100.0	11:01 (100)

Continued on next page ...

Table S1.2 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
11:04	110	0.0446	96	0.0399	78.1	98.7	62.7	100.0	100.0	98.8	11:01 (93)
12:01	35	0.0142	39	0.0162	87.2	99.9	94.1	100.0	97.0	99.9	11:01 (100)
12:02	2	0.0008	1	0.0004	100.0	100.0	100.0	100.0	100.0	100.0	–
13:01	170	0.0689	142	0.0591	95.1	99.8	98.5	99.9	98.5	99.9	13:02 (50)
13:02	89	0.0361	87	0.0362	90.8	100.0	100.0	100.0	98.8	100.0	–
13:03	30	0.0122	31	0.0129	93.5	100.0	100.0	100.0	100.0	100.0	–
14:01	68	0.0276	67	0.0279	94.0	99.8	98.4	99.8	93.9	100.0	15:01 (100)
15:01	294	0.1191	300	0.1248	96.0	99.8	99.7	99.8	99.0	100.0	03:01 (100)
15:02	28	0.0113	28	0.0116	96.4	100.0	100.0	100.0	100.0	100.0	–
16:01	68	0.0276	68	0.0283	92.6	99.9	100.0	99.9	96.9	100.0	–
16:02	7	0.0028	7	0.0029	85.7	100.0	83.3	100.0	100.0	100.0	16:01 (100)
HLA–DQA1: Overall accuracy: 97.8%											
01:01	202	0.1156	185	0.1068	99.5	99.7	97.3	100.0	100.0	99.7	01:04 (80)
01:02	351	0.2008	355	0.2050	98.0	99.8	99.7	99.9	99.4	99.9	01:03 (100)
01:03	147	0.0841	138	0.0797	95.7	99.8	98.5	99.9	99.2	99.9	01:02 (100)
01:04	30	0.0172	29	0.0167	96.6	99.7	100.0	99.7	84.8	100.0	–
01:05	6	0.0034	6	0.0035	66.7	99.9	100.0	99.9	88.9	100.0	–
02:01	237	0.1356	263	0.1518	98.9	100.0	100.0	100.0	100.0	100.0	–
03:01	164	0.0938	159	0.0918	98.1	98.6	94.2	99.1	91.3	99.4	03:03 (89)
03:02	14	0.0080	14	0.0081	85.7	100.0	100.0	100.0	100.0	100.0	–
03:03	108	0.0618	108	0.0624	98.1	98.7	86.8	99.5	91.5	99.1	03:01 (100)
04:01	54	0.0309	52	0.0300	94.2	99.8	100.0	99.8	94.2	100.0	–
05:01	194	0.1110	183	0.1057	100.0	100.0	100.0	100.0	100.0	100.0	–
05:05	227	0.1299	228	0.1316	98.7	99.8	100.0	99.7	98.3	100.0	–
06:01	5	0.0029	5	0.0029	80.0	99.9	50.0	100.0	100.0	99.9	04:01 (100)
HLA–DQB1 Overall accuracy: 99.2%											
02:01	220	0.1136	204	0.1067	99.0	100.0	100.0	100.0	100.0	100.0	–
02:02	200	0.1033	200	0.1046	98.0	99.9	100.0	99.9	99.5	100.0	–
03:01	393	0.2030	384	0.2008	98.7	99.7	99.7	99.7	98.8	99.9	02:02 (100)
03:02	187	0.0966	187	0.0978	98.9	99.9	100.0	99.9	98.9	100.0	–
03:03	84	0.0434	83	0.0434	94.0	99.9	98.7	100.0	100.0	99.9	03:01 (100)
03:04	5	0.0026	5	0.0026	60.0	99.9	33.3	100.0	100.0	99.9	03:01 (100)
03:19	2	0.0010	2	0.0010	100.0	99.9	50.0	100.0	100.0	99.9	03:01 (100)
04:02	61	0.0315	60	0.0314	98.3	100.0	100.0	100.0	100.0	100.0	–
05:01	220	0.1136	213	0.1114	98.6	99.9	99.0	100.0	100.0	99.9	05:03 (100)
05:02	67	0.0346	65	0.0340	98.5	100.0	100.0	100.0	100.0	100.0	–
05:03	55	0.0284	56	0.0293	100.0	99.9	100.0	99.9	96.6	100.0	–
06:01	24	0.0124	22	0.0115	95.5	100.0	100.0	100.0	100.0	100.0	–
06:02	234	0.1209	228	0.1192	97.8	99.7	99.6	99.8	98.4	99.9	06:04 (100)
06:03	112	0.0579	132	0.0690	97.0	99.8	97.7	100.0	100.0	99.8	06:02 (100)
06:04	52	0.0269	55	0.0288	94.5	99.8	98.1	99.9	96.2	99.9	03:01 (50)
06:09	14	0.0072	13	0.0068	92.3	99.9	91.7	100.0	100.0	99.9	06:04 (100)
HLA–DPB1: Overall accuracy: 94.8%											
01:01	74	0.0451	77	0.0479	96.1	100.0	100.0	100.0	100.0	100.0	–
02:01	201	0.1226	233	0.1449	96.6	98.0	90.7	99.2	95.3	98.5	04:01 (95)
03:01	133	0.0811	131	0.0815	93.9	98.1	99.2	98.0	80.8	99.9	104:01 (100)
04:01	713	0.4348	678	0.4216	98.5	97.8	99.7	96.3	95.3	99.8	02:01 (100)
04:02	212	0.1293	201	0.1250	97.0	99.5	100.0	99.5	96.8	100.0	–
05:01	33	0.0201	32	0.0199	100.0	100.0	100.0	100.0	100.0	100.0	–
09:01	12	0.0073	10	0.0062	50.0	99.9	100.0	99.9	83.3	100.0	–
10:01	22	0.0134	24	0.0149	91.7	99.9	95.5	100.0	100.0	99.9	09:01 (100)
11:01	29	0.0177	25	0.0155	88.0	100.0	100.0	100.0	100.0	100.0	–
13:01	38	0.0232	33	0.0205	87.9	100.0	100.0	100.0	100.0	100.0	–
14:01	20	0.0122	21	0.0131	66.7	100.0	100.0	100.0	100.0	100.0	–
15:01	16	0.0098	15	0.0093	100.0	100.0	100.0	100.0	100.0	100.0	–
16:01	7	0.0043	6	0.0037	100.0	100.0	100.0	100.0	100.0	100.0	–
17:01	29	0.0177	31	0.0193	93.5	100.0	100.0	100.0	100.0	100.0	–
19:01	8	0.0049	8	0.0050	100.0	100.0	100.0	100.0	100.0	100.0	–

Continued on next page ...

Table S1.2 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
104:01	24	0.0146	24	0.0149	87.5	99.9	100.0	99.9	91.3	100.0	–

[1]: the HLA alleles with more than one copy and non-zero sensitivity in the training are listed.
[2]: CR – call rate.
[3]: ACC – allele accuracy.
[4]: the most likely miscalled allele and the proportion of the most likely miscalled allele in all miscalled alleles.

Table S1.3: The sensitivity (SEN), specificity (SPE), positive predictive value (PPV) and negative predictive value (NPV) calculated from validation samples for each four-digit HLA allele with call threshold 0.5, when study data of Asian ancestry were divided to training and validation parts with approximately equal sizes. The SNP markers in the intersect of Illumina platforms were used.

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
HLA–A: Overall accuracy: 93.8%											
01:01	21	0.0331	21	0.0363	95.2	100.0	100.0	100.0	100.0	100.0	–
02:01	81	0.1278	65	0.1125	84.6	97.4	94.5	97.7	82.5	99.4	02:07 (67)
02:05	2	0.0032	2	0.0035	100.0	100.0	100.0	100.0	100.0	100.0	–
02:06	39	0.0615	27	0.0467	92.6	99.6	100.0	99.6	92.6	100.0	–
02:07	31	0.0489	32	0.0554	81.2	99.2	92.3	99.6	92.3	99.6	02:01 (100)
02:11	9	0.0142	8	0.0138	87.5	100.0	100.0	100.0	100.0	100.0	–
03:01	16	0.0252	13	0.0225	100.0	100.0	100.0	100.0	100.0	100.0	–
03:02	3	0.0047	2	0.0035	50.0	100.0	100.0	100.0	100.0	100.0	–
11:01	112	0.1767	120	0.2076	92.5	99.6	99.1	99.8	99.1	99.8	11:02 (100)
11:02	10	0.0158	7	0.0121	85.7	99.8	100.0	99.8	85.7	100.0	–
24:02	102	0.1609	103	0.1782	94.2	97.9	100.0	97.5	89.8	100.0	–
26:01	23	0.0363	17	0.0294	100.0	99.2	100.0	99.2	81.0	100.0	–
29:01	2	0.0032	1	0.0017	100.0	100.0	100.0	100.0	100.0	100.0	–
30:01	22	0.0347	20	0.0346	100.0	100.0	100.0	100.0	100.0	100.0	–
30:04	4	0.0063	3	0.0052	100.0	100.0	100.0	100.0	100.0	100.0	–
31:01	24	0.0379	21	0.0363	100.0	100.0	100.0	100.0	100.0	100.0	–
32:01	8	0.0126	9	0.0156	88.9	100.0	100.0	100.0	100.0	100.0	–
33:03	58	0.0915	59	0.1021	98.3	100.0	100.0	100.0	100.0	100.0	–
34:01	9	0.0142	7	0.0121	85.7	99.8	100.0	99.8	85.7	100.0	–
68:01	10	0.0158	7	0.0121	85.7	100.0	100.0	100.0	100.0	100.0	–
HLA–B: Overall accuracy: 94.7%											
07:02	22	0.0291	16	0.0239	93.8	100.0	100.0	100.0	100.0	100.0	–
07:05	12	0.0159	9	0.0134	88.9	100.0	100.0	100.0	100.0	100.0	–
08:01	11	0.0146	11	0.0164	90.9	100.0	100.0	100.0	100.0	100.0	–
13:01	27	0.0357	22	0.0328	77.3	100.0	100.0	100.0	100.0	100.0	–
13:02	27	0.0357	26	0.0388	84.6	100.0	100.0	100.0	100.0	100.0	–
14:01	4	0.0053	3	0.0045	66.7	100.0	100.0	100.0	100.0	100.0	–
15:01	24	0.0317	28	0.0418	46.4	99.0	92.3	99.2	75.0	99.8	46:01 (100)
15:02	22	0.0291	19	0.0284	84.2	99.8	93.8	100.0	100.0	99.8	15:35 (100)
15:17	3	0.0040	4	0.0060	50.0	100.0	100.0	100.0	100.0	100.0	–
15:18	10	0.0132	9	0.0134	77.8	100.0	100.0	100.0	100.0	100.0	–
15:35	6	0.0079	3	0.0045	100.0	99.8	100.0	99.8	75.0	100.0	–
18:01	8	0.0106	7	0.0104	42.9	100.0	100.0	100.0	100.0	100.0	–
27:04	8	0.0106	4	0.0060	100.0	99.6	100.0	99.6	66.7	100.0	–
27:05	9	0.0119	9	0.0134	66.7	100.0	100.0	100.0	100.0	100.0	–
35:01	27	0.0357	24	0.0358	70.8	99.4	100.0	99.4	85.0	100.0	–
35:03	17	0.0225	11	0.0164	54.5	99.4	50.0	100.0	100.0	99.6	35:01 (100)
35:05	6	0.0079	6	0.0090	50.0	100.0	100.0	100.0	100.0	100.0	–
37:01	13	0.0172	10	0.0149	80.0	100.0	100.0	100.0	100.0	100.0	–
38:01	4	0.0053	3	0.0045	66.7	100.0	100.0	100.0	100.0	100.0	–
38:02	23	0.0304	17	0.0254	82.4	99.2	100.0	99.1	77.8	100.0	–
39:01	11	0.0146	13	0.0194	53.8	99.4	57.1	100.0	100.0	99.5	38:02 (100)
40:01	61	0.0807	48	0.0716	85.4	100.0	100.0	100.0	100.0	100.0	–
40:02	22	0.0291	23	0.0343	65.2	99.2	86.7	99.6	86.7	99.7	40:06 (100)
40:06	37	0.0489	34	0.0507	67.6	99.0	91.3	99.3	87.5	99.7	27:04 (100)
44:02	7	0.0093	6	0.0090	66.7	100.0	100.0	100.0	100.0	100.0	–
44:03	29	0.0384	29	0.0433	89.7	100.0	100.0	100.0	100.0	100.0	–
46:01	46	0.0608	39	0.0582	84.6	99.6	97.0	99.8	97.0	99.8	40:02 (100)
47:01	2	0.0026	1	0.0015	100.0	100.0	100.0	100.0	100.0	100.0	–
48:01	13	0.0172	14	0.0209	64.3	99.8	100.0	99.8	90.0	100.0	–

Continued on next page ...

Table S1.3 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
50:01	4	0.0053	5	0.0075	80.0	100.0	100.0	100.0	100.0	100.0	–
51:01	50	0.0661	56	0.0836	83.9	99.2	95.7	99.5	95.7	99.7	51:02 (100)
51:02	4	0.0053	4	0.0060	75.0	99.4	66.7	99.6	50.0	99.8	51:01 (100)
51:06	3	0.0040	4	0.0060	75.0	100.0	100.0	100.0	100.0	100.0	–
52:01	31	0.0410	22	0.0328	68.2	99.8	93.3	100.0	100.0	99.8	51:01 (100)
54:01	27	0.0357	22	0.0328	72.7	99.4	93.8	99.6	88.2	99.8	55:02 (100)
57:01	10	0.0132	9	0.0134	88.9	100.0	100.0	100.0	100.0	100.0	–
58:01	35	0.0463	41	0.0612	90.2	100.0	100.0	100.0	100.0	100.0	–
67:01	4	0.0053	3	0.0045	66.7	99.8	50.0	100.0	100.0	99.9	38:02 (100)
HLA–C: Overall accuracy: 97.8%											
01:02	83	0.1305	73	0.1246	95.9	99.1	98.6	99.2	94.5	99.8	04:01 (50)
02:02	5	0.0079	4	0.0068	100.0	100.0	100.0	100.0	100.0	100.0	–
03:02	33	0.0519	29	0.0495	96.6	100.0	100.0	100.0	100.0	100.0	–
03:03	50	0.0786	38	0.0648	97.4	99.5	97.3	99.6	94.7	99.8	03:04 (100)
03:04	50	0.0786	53	0.0904	98.1	99.5	96.2	99.8	98.0	99.6	03:03 (100)
04:01	37	0.0582	32	0.0546	87.5	99.8	100.0	99.8	96.6	100.0	–
04:03	6	0.0094	5	0.0085	80.0	99.8	100.0	99.8	80.0	100.0	–
05:01	6	0.0094	6	0.0102	83.3	100.0	100.0	100.0	100.0	100.0	–
06:02	46	0.0723	45	0.0768	97.8	100.0	100.0	100.0	100.0	100.0	–
07:01	17	0.0267	14	0.0239	92.9	100.0	100.0	100.0	100.0	100.0	–
07:02	97	0.1525	104	0.1775	97.1	99.8	99.0	100.0	100.0	99.8	04:01 (50)
07:04	7	0.0110	6	0.0102	66.7	100.0	100.0	100.0	100.0	100.0	–
08:01	53	0.0833	50	0.0853	90.0	99.6	100.0	99.6	95.7	100.0	–
08:02	5	0.0079	5	0.0085	80.0	100.0	100.0	100.0	100.0	100.0	–
12:02	26	0.0409	25	0.0427	88.0	100.0	100.0	100.0	100.0	100.0	–
12:03	14	0.0220	11	0.0188	90.9	100.0	100.0	100.0	100.0	100.0	–
12:04	3	0.0047	3	0.0051	100.0	100.0	100.0	100.0	100.0	100.0	–
14:02	35	0.0550	32	0.0546	96.9	100.0	100.0	100.0	100.0	100.0	–
14:03	12	0.0189	13	0.0222	92.3	100.0	100.0	100.0	100.0	100.0	–
15:02	30	0.0472	30	0.0512	93.3	99.8	100.0	99.8	96.6	100.0	–
HLA–DRB1: Overall accuracy: 95.8%											
01:01	12	0.0165	14	0.0210	64.3	100.0	100.0	100.0	100.0	100.0	–
03:01	40	0.0551	34	0.0511	82.4	100.0	100.0	100.0	100.0	100.0	–
04:01	7	0.0096	7	0.0105	57.1	100.0	100.0	100.0	100.0	100.0	–
04:03	21	0.0289	20	0.0300	10.0	99.8	100.0	99.8	66.7	100.0	–
04:04	7	0.0096	6	0.0090	16.7	100.0	100.0	100.0	100.0	100.0	–
04:05	42	0.0579	34	0.0511	70.6	99.6	91.7	100.0	100.0	99.7	04:10 (50)
04:06	17	0.0234	13	0.0195	30.8	100.0	100.0	100.0	100.0	100.0	–
04:10	5	0.0069	4	0.0060	25.0	99.8	100.0	99.8	50.0	100.0	–
07:01	65	0.0895	61	0.0916	83.6	100.0	100.0	100.0	100.0	100.0	–
08:02	10	0.0138	7	0.0105	85.7	99.6	100.0	99.6	75.0	100.0	–
08:03	36	0.0496	36	0.0541	69.4	99.8	100.0	99.8	96.2	100.0	–
09:01	86	0.1185	65	0.0976	90.8	100.0	100.0	100.0	100.0	100.0	–
10:01	16	0.0220	18	0.0270	66.7	100.0	100.0	100.0	100.0	100.0	–
11:01	35	0.0482	27	0.0405	70.4	99.1	100.0	99.1	82.6	100.0	–
12:01	22	0.0303	26	0.0390	57.7	99.4	80.0	100.0	100.0	99.5	11:01 (67)
12:02	40	0.0551	53	0.0796	88.7	99.8	100.0	99.8	97.9	100.0	–
13:01	16	0.0220	13	0.0195	69.2	100.0	100.0	100.0	100.0	100.0	–
13:02	31	0.0427	28	0.0420	78.6	100.0	100.0	100.0	100.0	100.0	–
13:12	3	0.0041	2	0.0030	50.0	100.0	100.0	100.0	100.0	100.0	–
14:01	12	0.0165	15	0.0225	20.0	99.4	33.3	99.8	50.0	99.7	14:04 (100)
14:03	6	0.0083	5	0.0075	100.0	99.8	80.0	100.0	100.0	99.8	11:01 (100)
14:04	15	0.0207	14	0.0210	64.3	99.4	100.0	99.3	75.0	100.0	–
14:05	15	0.0207	15	0.0225	46.7	99.8	85.7	100.0	100.0	99.8	14:04 (100)
15:01	70	0.0964	66	0.0991	77.3	99.8	98.0	100.0	100.0	99.8	16:02 (100)
15:02	48	0.0661	50	0.0751	82.0	100.0	100.0	100.0	100.0	100.0	–
16:02	12	0.0165	13	0.0195	46.2	99.8	100.0	99.8	85.7	100.0	–
HLA–DQA1: Overall accuracy: 90.0%											

Continued on next page ...

Table S1.3 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
01:01	34	0.0570	31	0.0542	83.9	98.7	76.9	100.0	100.0	98.9	01:04 (83)
01:02	109	0.1829	106	0.1853	85.8	98.9	96.7	99.5	97.8	99.4	01:03 (67)
01:03	73	0.1225	77	0.1346	87.0	99.4	98.5	99.5	97.1	99.8	01:02 (100)
01:04	28	0.0470	24	0.0420	58.3	98.9	100.0	98.9	73.7	100.0	–
01:05	8	0.0134	8	0.0140	75.0	100.0	100.0	100.0	100.0	100.0	–
02:01	54	0.0906	55	0.0962	94.5	100.0	100.0	100.0	100.0	100.0	–
03:01	83	0.1393	68	0.1189	76.5	95.3	61.5	99.5	94.1	96.3	03:02 (56)
03:02	50	0.0839	55	0.0962	65.5	97.4	97.2	97.4	75.7	99.8	03:03 (100)
03:03	28	0.0470	27	0.0472	59.3	97.4	87.5	97.8	58.9	99.6	03:01 (100)
04:01	9	0.0151	7	0.0122	100.0	100.0	100.0	100.0	100.0	100.0	–
05:01	34	0.0570	27	0.0472	92.6	98.9	80.0	100.0	100.0	99.1	05:05 (80)
05:03	6	0.0101	5	0.0087	20.0	99.8	100.0	99.8	50.0	100.0	–
05:05	29	0.0487	28	0.0490	85.7	98.7	100.0	98.6	80.0	100.0	–
05:08	7	0.0117	6	0.0105	83.3	99.8	80.0	100.0	100.0	99.8	05:05 (100)
06:01	39	0.0654	45	0.0787	91.1	99.8	100.0	99.8	97.6	100.0	–
HLA–DQB1 Overall accuracy: 98.1%											
02:01	25	0.0399	27	0.0452	100.0	100.0	100.0	100.0	100.0	100.0	–
02:02	44	0.0703	38	0.0635	100.0	99.8	97.4	100.0	100.0	99.8	03:03 (100)
03:01	111	0.1773	104	0.1739	99.0	98.6	98.1	98.7	94.4	99.6	03:03 (100)
03:02	45	0.0719	50	0.0836	92.0	100.0	100.0	100.0	100.0	100.0	–
03:03	83	0.1326	86	0.1438	96.5	99.0	96.4	99.4	96.4	99.4	03:01 (100)
04:01	32	0.0511	26	0.0435	96.2	99.7	96.0	99.8	96.0	99.8	04:02 (100)
04:02	13	0.0208	12	0.0201	100.0	99.7	91.7	99.8	91.7	99.8	04:01 (100)
05:01	36	0.0575	34	0.0569	100.0	100.0	100.0	100.0	100.0	100.0	–
05:02	44	0.0703	38	0.0635	100.0	99.5	92.1	100.0	100.0	99.5	03:01 (100)
05:03	31	0.0495	35	0.0585	100.0	100.0	100.0	100.0	100.0	100.0	–
06:01	74	0.1182	78	0.1304	94.9	100.0	100.0	100.0	100.0	100.0	–
06:02	43	0.0687	34	0.0569	97.1	100.0	100.0	100.0	100.0	100.0	–
06:03	12	0.0192	12	0.0201	58.3	100.0	100.0	100.0	100.0	100.0	–
06:04	18	0.0288	13	0.0217	84.6	100.0	100.0	100.0	100.0	100.0	–
06:09	11	0.0176	11	0.0184	90.9	100.0	100.0	100.0	100.0	100.0	–
HLA–DPB1: Overall accuracy: 95.3%											
01:01	18	0.0332	15	0.0293	80.0	99.8	91.7	100.0	100.0	99.8	26:01 (100)
02:01	109	0.2011	113	0.2207	85.0	97.6	96.9	97.9	93.0	99.3	04:01 (67)
02:02	21	0.0387	24	0.0469	66.7	98.3	62.5	99.8	90.9	98.8	02:01 (100)
03:01	16	0.0295	13	0.0254	30.8	99.5	100.0	99.5	66.7	100.0	–
04:01	78	0.1439	66	0.1289	93.9	99.1	96.8	99.4	96.8	99.6	02:01 (50)
04:02	25	0.0461	29	0.0566	93.1	99.3	100.0	99.2	90.0	100.0	–
05:01	165	0.3044	154	0.3008	88.3	99.3	99.3	99.3	98.5	99.7	14:01 (100)
09:01	10	0.0185	12	0.0234	91.7	100.0	100.0	100.0	100.0	100.0	–
09:02	10	0.0185	8	0.0156	62.5	99.3	60.0	99.8	75.0	99.6	05:01 (50)
13:01	21	0.0387	20	0.0391	80.0	99.5	93.8	99.8	93.8	99.8	09:02 (100)
14:01	16	0.0295	18	0.0352	77.8	99.8	100.0	99.8	93.3	100.0	–
17:01	15	0.0277	14	0.0273	92.9	100.0	100.0	100.0	100.0	100.0	–
26:01	7	0.0129	6	0.0117	50.0	99.8	100.0	99.8	75.0	100.0	–
104:01	11	0.0203	9	0.0176	44.4	99.5	50.0	100.0	100.0	99.6	03:01 (100)

[1]: the HLA alleles with more than one copy and non-zero sensitivity in the training are listed.
[2]: CR – call rate.
[3]: ACC – allele accuracy.
[4]: the most likely miscalled allele and the proportion of the most likely miscalled allele in all miscalled alleles.

Table S1.4: The sensitivity (SEN), specificity (SPE), positive predictive value (PPV) and negative predictive value (NPV) calculated from validation samples for each four-digit HLA allele with call threshold 0.5, when study data of Hispanic ancestry were divided to training and validation parts with approximately equal sizes. The SNP markers in the intersect of Illumina platforms were used.

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
HLA–A: Overall accuracy: 96.0%											
01:01	22	0.0683	17	0.0620	100.0	100.0	100.0	100.0	100.0	100.0	–
02:01	78	0.2422	75	0.2737	84.0	98.2	100.0	97.5	94.0	100.0	–
02:05	2	0.0062	2	0.0073	50.0	100.0	100.0	100.0	100.0	100.0	–
02:06	4	0.0124	5	0.0182	40.0	100.0	100.0	100.0	100.0	100.0	–
03:01	21	0.0652	15	0.0547	100.0	100.0	100.0	100.0	100.0	100.0	–
11:01	14	0.0435	12	0.0438	83.3	100.0	100.0	100.0	100.0	100.0	–
23:01	13	0.0404	9	0.0328	88.9	100.0	100.0	100.0	100.0	100.0	–
24:02	33	0.1025	26	0.0949	80.8	100.0	100.0	100.0	100.0	100.0	–
26:01	13	0.0404	13	0.0474	84.6	99.1	100.0	99.1	84.6	100.0	–
29:02	17	0.0528	14	0.0511	100.0	100.0	100.0	100.0	100.0	100.0	–
30:01	6	0.0186	4	0.0146	75.0	100.0	100.0	100.0	100.0	100.0	–
30:02	5	0.0155	5	0.0182	60.0	100.0	100.0	100.0	100.0	100.0	–
30:04	2	0.0062	2	0.0073	100.0	100.0	100.0	100.0	100.0	100.0	–
31:01	20	0.0621	24	0.0876	91.7	100.0	100.0	100.0	100.0	100.0	–
32:01	6	0.0186	5	0.0182	80.0	100.0	100.0	100.0	100.0	100.0	–
33:01	4	0.0124	4	0.0146	75.0	100.0	100.0	100.0	100.0	100.0	–
68:01	18	0.0559	20	0.0730	70.0	98.7	92.9	99.1	86.7	99.6	68:17 (100)
68:02	4	0.0124	3	0.0109	100.0	100.0	100.0	100.0	100.0	100.0	–
68:17	4	0.0124	4	0.0146	25.0	99.6	100.0	99.6	50.0	100.0	–
HLA–B: Overall accuracy: 93.8%											
07:02	14	0.0294	16	0.0417	68.8	100.0	100.0	100.0	100.0	100.0	–
08:01	17	0.0357	12	0.0312	91.7	100.0	100.0	100.0	100.0	100.0	–
13:02	5	0.0105	10	0.0260	40.0	100.0	100.0	100.0	100.0	100.0	–
14:01	4	0.0084	3	0.0078	66.7	100.0	100.0	100.0	100.0	100.0	–
14:02	7	0.0147	9	0.0234	44.4	100.0	100.0	100.0	100.0	100.0	–
15:01	14	0.0294	9	0.0234	33.3	98.6	66.7	99.3	66.7	99.7	35:43 (100)
15:04	12	0.0252	14	0.0365	35.7	100.0	100.0	100.0	100.0	100.0	–
18:01	20	0.0420	15	0.0391	46.7	100.0	100.0	100.0	100.0	100.0	–
27:05	5	0.0105	3	0.0078	66.7	100.0	100.0	100.0	100.0	100.0	–
35:02	5	0.0105	4	0.0104	75.0	100.0	100.0	100.0	100.0	100.0	–
35:05	11	0.0231	11	0.0286	18.2	100.0	100.0	100.0	100.0	100.0	–
35:19	4	0.0084	2	0.0052	50.0	100.0	100.0	100.0	100.0	100.0	–
35:43	8	0.0168	5	0.0130	40.0	98.6	100.0	98.6	50.0	100.0	–
37:01	2	0.0042	3	0.0078	33.3	100.0	100.0	100.0	100.0	100.0	–
38:01	9	0.0189	7	0.0182	57.1	100.0	100.0	100.0	100.0	100.0	–
39:06	7	0.0147	3	0.0078	33.3	100.0	100.0	100.0	100.0	100.0	–
39:09	11	0.0231	6	0.0156	50.0	100.0	100.0	100.0	100.0	100.0	–
40:01	9	0.0189	5	0.0130	20.0	99.3	100.0	99.3	50.0	100.0	–
41:01	3	0.0063	2	0.0052	50.0	100.0	100.0	100.0	100.0	100.0	–
42:01	2	0.0042	1	0.0026	100.0	100.0	100.0	100.0	100.0	100.0	–
44:02	10	0.0210	12	0.0312	50.0	99.3	83.3	100.0	100.0	99.7	40:01 (100)
44:03	36	0.0756	22	0.0573	77.3	100.0	100.0	100.0	100.0	100.0	–
48:01	19	0.0399	16	0.0417	50.0	100.0	100.0	100.0	100.0	100.0	–
49:01	11	0.0231	11	0.0286	54.5	100.0	100.0	100.0	100.0	100.0	–
50:01	5	0.0105	5	0.0130	60.0	100.0	100.0	100.0	100.0	100.0	–

Continued on next page ...

Table S1.4 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
51:01	37	0.0777	27	0.0703	48.1	99.3	100.0	99.2	92.9	100.0	–
52:01	11	0.0231	11	0.0286	72.7	99.3	87.5	100.0	100.0	99.7	35:01 (50)
53:01	4	0.0084	2	0.0052	50.0	100.0	100.0	100.0	100.0	100.0	–
57:01	8	0.0168	4	0.0104	100.0	100.0	100.0	100.0	100.0	100.0	–
58:01	5	0.0105	6	0.0156	50.0	100.0	100.0	100.0	100.0	100.0	–
HLA–C: Overall accuracy: 98.4%											
01:02	24	0.0764	25	0.0874	80.0	99.6	95.0	100.0	100.0	99.6	08:01 (100)
02:02	7	0.0223	6	0.0210	50.0	100.0	100.0	100.0	100.0	100.0	–
03:03	10	0.0318	5	0.0175	100.0	100.0	100.0	100.0	100.0	100.0	–
03:04	25	0.0796	26	0.0909	76.9	98.8	100.0	98.7	87.0	100.0	–
04:01	52	0.1656	49	0.1713	95.9	100.0	100.0	100.0	100.0	100.0	–
05:01	13	0.0414	12	0.0420	66.7	100.0	100.0	100.0	100.0	100.0	–
06:02	19	0.0605	17	0.0594	94.1	100.0	100.0	100.0	100.0	100.0	–
07:01	31	0.0987	29	0.1014	100.0	100.0	100.0	100.0	100.0	100.0	–
07:02	34	0.1083	30	0.1049	96.7	100.0	100.0	100.0	100.0	100.0	–
08:01	5	0.0159	5	0.0175	80.0	99.6	100.0	99.6	80.0	100.0	–
08:02	8	0.0255	8	0.0280	100.0	100.0	100.0	100.0	100.0	100.0	–
08:03	9	0.0287	7	0.0245	85.7	100.0	100.0	100.0	100.0	100.0	–
12:02	5	0.0159	5	0.0175	100.0	100.0	100.0	100.0	100.0	100.0	–
12:03	16	0.0510	12	0.0420	100.0	100.0	100.0	100.0	100.0	100.0	–
14:02	6	0.0191	6	0.0210	66.7	100.0	100.0	100.0	100.0	100.0	–
15:02	14	0.0446	15	0.0524	86.7	100.0	100.0	100.0	100.0	100.0	–
16:01	14	0.0446	14	0.0490	100.0	100.0	100.0	100.0	100.0	100.0	–
16:02	4	0.0127	3	0.0105	100.0	100.0	100.0	100.0	100.0	100.0	–
17:01	3	0.0096	4	0.0140	25.0	100.0	100.0	100.0	100.0	100.0	–
HLA–DRB1: Overall accuracy: 93.5%											
01:01	12	0.0269	12	0.0305	41.7	99.5	80.0	100.0	100.0	99.7	01:03 (100)
01:02	8	0.0179	9	0.0228	100.0	100.0	100.0	100.0	100.0	100.0	–
03:01	28	0.0628	27	0.0685	70.4	100.0	100.0	100.0	100.0	100.0	–
04:01	8	0.0179	6	0.0152	33.3	99.5	50.0	100.0	100.0	99.7	07:01 (50)
04:04	24	0.0538	16	0.0406	18.8	99.0	33.3	100.0	100.0	99.5	04:07 (100)
04:05	4	0.0090	4	0.0102	50.0	100.0	100.0	100.0	100.0	100.0	–
04:06	2	0.0045	2	0.0051	50.0	100.0	100.0	100.0	100.0	100.0	–
04:07	41	0.0919	31	0.0787	22.6	98.5	100.0	98.4	70.0	100.0	–
07:01	46	0.1031	42	0.1066	83.3	99.5	100.0	99.4	97.2	100.0	–
08:01	5	0.0112	5	0.0127	60.0	99.5	66.7	100.0	100.0	99.7	08:02 (100)
08:02	37	0.0830	32	0.0812	62.5	98.0	100.0	97.8	83.3	100.0	–
09:01	26	0.0583	24	0.0609	75.0	99.5	100.0	99.5	94.7	100.0	–
10:01	4	0.0090	3	0.0076	100.0	100.0	100.0	100.0	100.0	100.0	–
11:04	17	0.0381	17	0.0431	5.9	100.0	100.0	100.0	100.0	100.0	–
12:01	6	0.0135	6	0.0152	50.0	100.0	100.0	100.0	100.0	100.0	–
13:01	14	0.0314	13	0.0330	69.2	100.0	100.0	100.0	100.0	100.0	–
13:02	16	0.0359	12	0.0305	50.0	100.0	100.0	100.0	100.0	100.0	–
14:01	6	0.0135	5	0.0127	80.0	100.0	100.0	100.0	100.0	100.0	–
14:02	22	0.0493	20	0.0508	55.0	98.5	81.8	99.5	90.0	99.5	14:06 (100)
14:06	9	0.0202	11	0.0279	63.6	98.5	85.7	99.0	75.0	99.7	07:01 (50)
15:01	20	0.0448	22	0.0558	68.2	100.0	100.0	100.0	100.0	100.0	–
15:02	6	0.0135	8	0.0203	87.5	100.0	100.0	100.0	100.0	100.0	–
16:01	3	0.0067	4	0.0102	100.0	100.0	100.0	100.0	100.0	100.0	–
16:02	9	0.0202	9	0.0228	11.1	100.0	100.0	100.0	100.0	100.0	–
HLA–DQA1: Overall accuracy: 95.8%											
01:01	20	0.0719	20	0.0769	100.0	98.3	100.0	98.1	83.3	100.0	–
01:02	28	0.1007	30	0.1154	83.3	98.7	92.0	99.5	95.8	99.2	01:01 (100)
01:03	14	0.0504	13	0.0500	92.3	99.6	91.7	100.0	100.0	99.6	01:02 (100)
01:04	3	0.0108	3	0.0115	66.7	99.6	50.0	100.0	100.0	99.6	01:01 (100)
02:01	37	0.1331	31	0.1192	100.0	100.0	100.0	100.0	100.0	100.0	–
03:01	57	0.2050	47	0.1808	89.4	97.9	100.0	97.4	89.4	100.0	–
03:02	15	0.0540	15	0.0577	86.7	99.2	84.6	100.0	100.0	99.2	03:01 (100)

Continued on next page ...

Table S1.4 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
03:03	11	0.0396	11	0.0423	72.7	98.7	62.5	100.0	100.0	98.8	03:01 (100)
04:01	33	0.1187	26	0.1000	100.0	100.0	100.0	100.0	100.0	100.0	–
05:01	21	0.0755	19	0.0731	100.0	100.0	100.0	100.0	100.0	100.0	–
05:03	11	0.0396	10	0.0385	90.0	100.0	100.0	100.0	100.0	100.0	–
05:05	26	0.0935	34	0.1308	82.4	100.0	100.0	100.0	100.0	100.0	–
HLA–DQB1 Overall accuracy: 98.9%											
02:01	23	0.0710	23	0.0767	91.3	100.0	100.0	100.0	100.0	100.0	–
02:02	36	0.1111	29	0.0967	96.6	99.6	96.4	100.0	100.0	99.6	03:01 (100)
03:01	62	0.1914	64	0.2133	89.1	98.9	100.0	98.6	95.0	100.0	–
03:02	62	0.1914	58	0.1933	96.6	100.0	100.0	100.0	100.0	100.0	–
03:03	20	0.0617	21	0.0700	76.2	100.0	100.0	100.0	100.0	100.0	–
04:02	39	0.1204	30	0.1000	90.0	100.0	100.0	100.0	100.0	100.0	–
05:01	27	0.0833	27	0.0900	100.0	100.0	100.0	100.0	100.0	100.0	–
05:03	4	0.0123	5	0.0167	60.0	100.0	100.0	100.0	100.0	100.0	–
06:01	5	0.0154	4	0.0133	100.0	100.0	100.0	100.0	100.0	100.0	–
06:02	19	0.0586	17	0.0567	82.4	100.0	100.0	100.0	100.0	100.0	–
06:03	11	0.0340	9	0.0300	100.0	100.0	100.0	100.0	100.0	100.0	–
06:04	7	0.0216	6	0.0200	100.0	100.0	100.0	100.0	100.0	100.0	–
HLA–DPB1: Overall accuracy: 97.5%											
01:01	10	0.0360	10	0.0403	100.0	100.0	100.0	100.0	100.0	100.0	–
02:01	31	0.1115	30	0.1210	70.0	99.0	95.2	99.4	95.2	99.6	04:01 (100)
02:02	3	0.0108	3	0.0121	66.7	100.0	100.0	100.0	100.0	100.0	–
04:01	70	0.2518	71	0.2863	87.3	99.5	100.0	99.3	98.4	100.0	–
04:02	75	0.2698	64	0.2581	96.9	100.0	100.0	100.0	100.0	100.0	–
05:01	10	0.0360	9	0.0363	88.9	100.0	100.0	100.0	100.0	100.0	–
11:01	8	0.0288	8	0.0323	75.0	100.0	100.0	100.0	100.0	100.0	–
13:01	13	0.0468	9	0.0363	77.8	99.5	100.0	99.5	87.5	100.0	–
14:01	18	0.0647	17	0.0685	88.2	99.5	100.0	99.5	93.8	100.0	–
17:01	5	0.0180	5	0.0202	40.0	100.0	100.0	100.0	100.0	100.0	–
104:01	4	0.0144	4	0.0161	75.0	100.0	100.0	100.0	100.0	100.0	–

[1]: the HLA alleles with more than one copy and non-zero sensitivity in the training are listed.
[2]: CR – call rate.
[3]: ACC – allele accuracy.
[4]: the most likely miscalled allele and the proportion of the most likely miscalled allele in all miscalled alleles.

Table S1.5: The sensitivity (SEN), specificity (SPE), positive predictive value (PPV) and negative predictive value (NPV) calculated from validation samples for each four-digit HLA allele with call threshold 0.5, when study data of African ancestry were divided to training and validation parts with approximately equal sizes. The SNP markers in the intersect of Illumina platforms were used.

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
HLA–A: Overall accuracy: 100%											
01:01	5	0.0309	4	0.0339	75.0	100.0	100.0	100.0	100.0	100.0	–
02:01	14	0.0864	10	0.0847	80.0	100.0	100.0	100.0	100.0	100.0	–
02:05	4	0.0247	2	0.0169	50.0	100.0	100.0	100.0	100.0	100.0	–
03:01	15	0.0926	9	0.0763	88.9	100.0	100.0	100.0	100.0	100.0	–
23:01	11	0.0679	11	0.0932	90.9	100.0	100.0	100.0	100.0	100.0	–
24:02	3	0.0185	2	0.0169	100.0	100.0	100.0	100.0	100.0	100.0	–
26:01	4	0.0247	3	0.0254	66.7	100.0	100.0	100.0	100.0	100.0	–
29:02	3	0.0185	4	0.0339	25.0	100.0	100.0	100.0	100.0	100.0	–
30:01	14	0.0864	11	0.0932	90.9	100.0	100.0	100.0	100.0	100.0	–
30:02	8	0.0494	11	0.0932	72.7	100.0	100.0	100.0	100.0	100.0	–
33:01	3	0.0185	2	0.0169	100.0	100.0	100.0	100.0	100.0	100.0	–
33:03	13	0.0802	6	0.0508	100.0	100.0	100.0	100.0	100.0	100.0	–
34:02	6	0.0370	6	0.0508	100.0	100.0	100.0	100.0	100.0	100.0	–
36:01	9	0.0556	12	0.1017	91.7	100.0	100.0	100.0	100.0	100.0	–
68:01	5	0.0309	4	0.0339	75.0	100.0	100.0	100.0	100.0	100.0	–
68:02	13	0.0802	10	0.0847	70.0	100.0	100.0	100.0	100.0	100.0	–
HLA–B: Overall accuracy: 96.7%											
07:02	17	0.0850	10	0.0704	30.0	100.0	100.0	100.0	100.0	100.0	–
14:02	2	0.0100	2	0.0141	50.0	100.0	100.0	100.0	100.0	100.0	–
15:03	9	0.0450	8	0.0563	25.0	100.0	100.0	100.0	100.0	100.0	–
18:01	8	0.0400	4	0.0282	25.0	100.0	100.0	100.0	100.0	100.0	–
35:01	14	0.0700	9	0.0634	22.2	100.0	100.0	100.0	100.0	100.0	–
42:01	11	0.0550	6	0.0423	50.0	96.7	100.0	96.3	75.0	100.0	–
45:01	6	0.0300	3	0.0211	100.0	100.0	100.0	100.0	100.0	100.0	–
49:01	6	0.0300	5	0.0352	20.0	100.0	100.0	100.0	100.0	100.0	–
53:01	24	0.1200	21	0.1479	47.6	100.0	100.0	100.0	100.0	100.0	–
57:03	8	0.0400	6	0.0423	16.7	100.0	100.0	100.0	100.0	100.0	–
58:02	8	0.0400	5	0.0352	40.0	100.0	100.0	100.0	100.0	100.0	–
HLA–C: Overall accuracy: 96.5%											
02:02	9	0.0608	9	0.0692	44.4	97.7	50.0	100.0	100.0	98.4	02:10 (100)
02:10	5	0.0338	5	0.0385	20.0	97.7	100.0	97.6	33.3	100.0	–
03:02	4	0.0270	5	0.0385	80.0	100.0	100.0	100.0	100.0	100.0	–
04:01	28	0.1892	25	0.1923	80.0	100.0	100.0	100.0	100.0	100.0	–
06:02	6	0.0405	11	0.0846	63.6	100.0	100.0	100.0	100.0	100.0	–
07:01	24	0.1622	9	0.0692	100.0	100.0	100.0	100.0	100.0	100.0	–
07:02	13	0.0878	8	0.0615	62.5	100.0	100.0	100.0	100.0	100.0	–
08:04	8	0.0541	4	0.0308	50.0	98.8	100.0	98.8	66.7	100.0	–
14:02	3	0.0203	3	0.0231	66.7	100.0	100.0	100.0	100.0	100.0	–
16:01	15	0.1014	20	0.1538	90.0	100.0	100.0	100.0	100.0	100.0	–
17:01	11	0.0743	12	0.0923	91.7	100.0	100.0	100.0	100.0	100.0	–
18:01	5	0.0338	3	0.0231	66.7	100.0	100.0	100.0	100.0	100.0	–
HLA–DRB1: Overall accuracy: 100%											
01:02	5	0.0281	3	0.0208	33.3	100.0	100.0	100.0	100.0	100.0	–
03:01	11	0.0618	9	0.0625	22.2	100.0	100.0	100.0	100.0	100.0	–
03:02	9	0.0506	11	0.0764	27.3	100.0	100.0	100.0	100.0	100.0	–
07:01	19	0.1067	13	0.0903	30.8	100.0	100.0	100.0	100.0	100.0	–
08:04	11	0.0618	11	0.0764	36.4	100.0	100.0	100.0	100.0	100.0	–
09:01	6	0.0337	6	0.0417	33.3	100.0	100.0	100.0	100.0	100.0	–
11:01	13	0.0730	12	0.0833	33.3	100.0	100.0	100.0	100.0	100.0	–
11:02	6	0.0337	5	0.0347	40.0	100.0	100.0	100.0	100.0	100.0	–

Continued on next page ...

Table S1.5 – continued from previous page

Allele[1]	Num. Train	Freq. Train	Num. Valid.	Freq. Valid.	CR[2] (%)	ACC[3] (%)	SEN (%)	SPE (%)	PPV (%)	NPV (%)	Miscall[4] (%)
13:01	13	0.0730	13	0.0903	15.4	100.0	100.0	100.0	100.0	100.0	–
13:02	10	0.0562	8	0.0556	12.5	100.0	100.0	100.0	100.0	100.0	–
13:03	11	0.0618	8	0.0556	25.0	100.0	100.0	100.0	100.0	100.0	–
15:03	17	0.0955	13	0.0903	30.8	100.0	100.0	100.0	100.0	100.0	–
16:02	3	0.0169	3	0.0208	33.3	100.0	100.0	100.0	100.0	100.0	–
HLA–DQA1: Overall accuracy: 97.2%											
01:01	10	0.0725	9	0.0692	11.1	100.0	100.0	100.0	100.0	100.0	–
01:02	41	0.2971	31	0.2385	35.5	100.0	100.0	100.0	100.0	100.0	–
01:03	10	0.0725	10	0.0769	40.0	100.0	100.0	100.0	100.0	100.0	–
02:01	16	0.1159	16	0.1231	56.2	100.0	100.0	100.0	100.0	100.0	–
04:01	15	0.1087	15	0.1154	33.3	100.0	100.0	100.0	100.0	100.0	–
05:01	14	0.1014	19	0.1462	26.3	97.2	100.0	96.8	83.3	100.0	–
HLA–DQB1 Overall accuracy: 97.7%											
02:01	17	0.1149	16	0.1270	18.8	100.0	100.0	100.0	100.0	100.0	–
03:01	20	0.1351	16	0.1270	25.0	100.0	100.0	100.0	100.0	100.0	–
03:02	7	0.0473	6	0.0476	66.7	100.0	100.0	100.0	100.0	100.0	–
04:02	11	0.0743	10	0.0794	70.0	100.0	100.0	100.0	100.0	100.0	–
05:01	18	0.1216	19	0.1508	47.4	97.7	100.0	97.1	90.0	100.0	–
05:02	6	0.0405	5	0.0397	60.0	100.0	100.0	100.0	100.0	100.0	–
06:02	26	0.1757	18	0.1429	66.7	97.7	91.7	100.0	100.0	99.1	05:01 (100)
06:03	7	0.0473	6	0.0476	16.7	100.0	100.0	100.0	100.0	100.0	–
06:09	4	0.0270	2	0.0159	50.0	100.0	100.0	100.0	100.0	100.0	–
HLA–DPB1: Overall accuracy: 75.0%											
01:01	17	0.1932	16	0.2581	31.2	75.0	100.0	33.3	71.4	100.0	–
04:01	8	0.0909	6	0.0968	16.7	100.0	100.0	100.0	100.0	100.0	–

[1]: the HLA alleles with more than one copy and non-zero sensitivity in the training are listed.
[2]: CR – call rate.
[3]: ACC – allele accuracy.
[4]: the most likely miscalled allele and the proportion of the most likely miscalled allele in all miscalled alleles.

VITA

Xiuwen Zheng was born on June 2nd, 1983 in Hainan, China. He received a Bachelor of Science from the University of Science and Technology of China in 2005, and a Master of Science in statistics from the University of Texas at Dallas in 2007. He earned a Doctor of Philosophy in biostatistics from the University of Washington in the year 2013.