A Flexible and Computationally Tractable Model for Patterns of Population Genetic Variation

Paul A. Scheet

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

Doctor of Philosophy

University of Washington

2006

Program Authorized to Offer Degree: Statistics
University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Paul A. Scheet

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

________________________________________
Matthew Stephens

Reading Committee:

________________________________________
Patrick Heagerty

________________________________________
Matthew Stephens

________________________________________
Elizabeth Thompson

Date: ________________________________
In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with “fair use” as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted “the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform.”

Signature________________________________________

Date____________________________________________
University of Washington

Abstract

A Flexible and Computationally Tractable Model for Patterns of Population Genetic Variation

Paul A. Scheet

Chair of the Supervisory Committee:
Associate Professor Matthew Stephens
Statistics

I present a statistical model for patterns of genetic variation in samples of unrelated individuals from natural populations. This model is based on the idea that, over short regions, haplotypes in a population tend to cluster into groups of similar haplotypes. To capture the fact that, due to recombination, this clustering tends to be local in nature, the model allows cluster memberships to change continuously along the chromosome according to a hidden Markov model. This approach is flexible, allowing for both “block-like” patterns of linkage disequilibrium (LD), and gradual decline in LD with distance. The algorithm for parameter estimation is fast, and as a result the model is practicable for large data sets (e.g. thousands of individuals typed at hundreds of thousands of markers). I apply the model to dense SNP data for the tasks of imputing missing genotypes and estimating haplotypic phase. The methods for genotype and haplotype estimation are competitive with the best available method but require a fraction of the computational effort. I extend the model to account explicitly for genotyping errors and present a method to detect errors that are clustered at specific sites in a dense scan of SNP genotypes. I consider a certain type of population structure and present an extension of the model to account for this structure, which offers improvement in genotype estimation in samples from geographically-separated
regions. Finally, I consider several constraints which impose local parametric structure, exploiting relationships among parameters expected in models for genetic data from real populations. The methods described herein are implemented in a software package, fastPHASE.
# TABLE OF CONTENTS

List of Figures ................................................................. iii

List of Tables ................................................................. vi

Chapter 1: Introduction ......................................................... 1
  1.1 Linkage Disequilibrium ................................................... 2
  1.2 Motivation for a Flexible and Computationally Tractable Model .. 8

Chapter 2: A Mixture Model for Haploids and Diploids .................... 9
  2.1 A Cluster Model for Haplotypes ......................................... 9
  2.2 Local Clustering of Haplotypes ...................................... 11
  2.3 Extension to Genotype Data .......................................... 12
  2.4 Parameter Estimation .................................................. 15
  2.5 Remarks ....................................................................... 17

Chapter 3: Inference of Missing Genotypes and Haplotypes ................ 18
  3.1 Review of Existing Methods ............................................ 18
  3.2 Missing Genotype Imputation .......................................... 19
  3.3 Haplotype Estimation .................................................... 21
  3.4 Results ....................................................................... 22
  3.5 Calibration of Predictive Probabilities ............................... 31
  3.6 Computational Requirements .......................................... 34
  3.7 Remarks ..................................................................... 34

Chapter 4: Imposing Parametric Structure ..................................... 37
  4.1 Parametric Constraints in the Markov Chain for Cluster Membership 37
  4.2 Estimation of Cluster-Specific Allele Frequencies at Untyped SNPs .. 41
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Dense genotype data. Genotypes from 23 African Americans and 24 CEPH individuals in the SeattleSNPs database are represented visually. Each individual's genotypes make up a row of the plot, with columns consisting of all the sampled genotypes at a SNP. Every colored square denotes a genotype: blue for the genotype consisting of two copies of the most common allele at a SNP (column); yellow for 2 copies of the less-common allele; red for one copy of each; and gray for missing genotypes. The image was created with the Visual Genotypes software on the SeattleSNPs website.</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Illustration of how the model allows cluster membership to change continuously along a chromosome. Each column represents a SNP, with the two alleles indicated by open and crossed squares. Successive pairs of rows represent the estimated pair of haplotypes for successive individuals. Colors represent estimated cluster membership of each allele, which changes as one moves along each haplotype. Locally each cluster can be thought of as representing a (common) combination of alleles at tightly-linked SNPs, and the figure illustrates how each haplotype is modelled as a mosaic of these common combinations. The figure was produced by fitting the model to the HapMap data from 60 unrelated CEPH individuals (Section 3.4.1), and then taking a single sample of cluster memberships and haplotypes from their conditional distribution, given the genotype data and parameter estimates (Appendix B). For brevity, haplotypes from only 10 individuals are shown.</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Calibration of the model for predicting uncertainty in inferred genotypes and haplotypes. Points (△) represent probabilities obtained by averaging over the 20 runs of the EM algorithm as described in the text.</td>
<td>33</td>
</tr>
</tbody>
</table>
4.1 *Representations of elements of $\theta$. Gray-scale representation of the values of $\theta$ for a 80-SNP region of CEU chromosome 22 of the HapMap. Values near one are black; values near zero are white. Cluster-specific allele frequencies are in columns; rows correspond to allele frequencies of clusters. TOP: $\theta$ values for SNP 10,000 to 10,080, order consistent with the data. BOTTOM: Columns in top plot are randomly permuted.*

6.1 *Empirical CDF of $\lambda$. Empirical distributions of $\lambda$ for SNPs with 0, 1, and at least 2 MIs, labeled with “0”, “1”, and “2”, respectively. Values for $\lambda$ are obtained from 5 random starts of EM, with $K = 5$. Sample sizes for each type of SNP are: 124 with at least 2 MIs, 150 SNPs with 1 MI, and 650 with 0 MIs. Often, when $\hat{e}_m$ is close to 0, setting $e_m$ to zero does not provide a real constraint and the small number of additional EM iterations will cause the value of $\lambda_m$ to be $\leq 0$. Only nonnegative values are plotted.*

6.2 *True positives vs. false positives for 3 error models.* True positives are sites with multiple MIs and false positives are sites with zero MIs. Values of $\lambda$ used to classify sites were obtained from 5 random starts of the EM algorithm with $K = 5$, calculated at all sites for which the expected number of genotype errors was at least one-half.

6.3 *True positives vs. false positives for chromosome 7.* Bottom plot corresponds to a zoomed-in region of the top plot. True and false positives were counted as for Figure 6.2, using $\lambda$ and the expected number of errors as the classification criteria. True positives are sites with multiple MIs. False positives are sites with zero MIs. The values of $\lambda$ used for classification were obtained from the 5 runs of the EM which produced the results in Figure 6.4. By only calculating $\lambda$ at sites with at least .5 expected errors, the number of sites considered was reduced to 5722.

6.4 *Expected number of errors and $\lambda$. The values are obtained from 5 starts of the EM algorithm. During each run of the EM, values of $\lambda$ were calculated at SNPs which had at least .5 expected errors. $K$ was set at 5. The $\circ$, $\ast$, and $\diamond$ symbols represent SNPs with 0, 1, and at least 2 MIs, respectively. The $\ast$ symbol denotes SNPs with genotypes discordant from other genotypes which did pass QC. The point circled in blue corresponds to the SNP genotypes in Figure 6.6.*
6.5 Calibration of expected error. For each value of observed MI, distributions of expected number of errors in SNPs which had exactly this number of MIs are plotted. Only SNPs with significant signal for error ($\lambda > 2.7$) are included in the calculations of expected number of errors.  

6.6 Genotype intensities of HapMap data. The axes are genotype intensities from the Illumina genotyping platform for the two alleles. LEFT. Example of a normal intensity plot from the HapMap. In the left plot, the 3 genotype classes are easily visible as clusters of genotypes. The middle cluster at approximately 45 degrees from the origin contains the heterozygote genotypes. RIGHT. Example of intensities which confirm a SNP suspected to contain genotype errors from the HapMap. Here, the three clusters may be an indication of increasing copies of a second allele (that is, they all have one common allele corresponding to the vertical position); or they perhaps should be called as homozygotes, heterozygotes, and homozygotes (minor). Instead the two right-most clusters were grouped together and called as heterozygotes.  

6.7 Genotype intensities of PARC data. The vertical axes are for genotype intensities from the Illumina genotyping platform. Horizontal axes display the angle of the points in corresponding plots with “Cartesian” coordinates (these are a transformation of the types of plots in Figure 6.6). LEFT. Example of a normal intensity plot from the PARC study. The 3 genotype classes cluster into 3 groups. RIGHT. Example of intensities which confirm a SNP suspected to contain genotype errors in the PARC data. This plot contains three clusters; however, the “middle” cluster contains a mix of red and pink colored genotypes, indicating that the red genotypes in that cluster were miscalled as homozygotes instead of heterozygotes.
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td><em>Summary of haplotype inference algorithms for unrelated individuals.</em></td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td><em>Error rates for estimating missing genotypes: SeattleSNPs.</em> Error rates are based on estimating 9479 missing genotypes. AA = Error rate on African American sample, ED = Error rate on European Descent sample, Total = Error rate on combined sample. The best-performing method in each column is highlighted in bold. Differences among fastPHASE, PHASE, and HaploBlock are not statistically significant; differences between these and the other methods are significant (p-value &lt; .007 for contrasts with GERBIL) based on bootstrap resampling of the 50 genes. For fastPHASE, K was selected independently for each analysis (AA only, ED only, and combined).*</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td><em>Error rates for estimating missing genotypes: CEPH HapMap data.</em> For each chromosome, 10% and 25% of the data were masked, resulting in 242,481 and 606,985 missing genotypes for chromosome 7, and 93,476 and 232,731 missing genotypes for chromosome 22. Results for fastPHASE were obtained from 20 random starts of the EM algorithm, except for the “1 start” case, where results were obtained from a single random start.*</td>
<td>28</td>
</tr>
<tr>
<td>3.4</td>
<td><em>Error rates for estimating missing genotypes with fastPHASE: CEPH HapMap data, chromosome 22.</em> I masked 10-80% of the data at random, resulting in 93,476 to 837,853 missing genotypes, and applied fastPHASE to the entire chromosome.*</td>
<td>28</td>
</tr>
</tbody>
</table>
3.5 Individual and switch error rates for haplotype estimates produced by different methods for the X-chromosome data. The best-performing method in each column is highlighted in bold. The results for Haplo-typer are taken from Lin et al. (2002). PL-EM and snphap failed to complete runs for the majority of simulations. For fastPHASE, PHASE (default), GERBIL and HaploBlock, the proportions (error rates) are computed by summing the numerator and denominator over all $8 \times 100$ datasets; whereas for all the others, proportions are an average of the error rates for each of the 8 genes. For fastPHASE, $K$ was chosen using a subset of the data and then fixed for all 800 data sets. *Computed with only partial data. ........................................ 30

3.6 Accuracy of haplotype inference methods for simulated data (Trials 1, 2 and 4) and real data from the HapMap project. Results for fastPHASE were obtained with the $(\hat{\delta}_w)$ estimator. Results for wphase (N Patterson, personal communication), HAP and hap2 were obtained by Marchini et al. (2006). Ind = individual error rate; Switch = switch error rate; Imp = genotype imputation error rate. .................. 32

3.7 Comparison of computation times (in hours) for different methods. Each method was applied on a 3 GHz Xeon processor with 1 GB of memory. The SeattleSNPs column is for analysis of all 47 individuals together. For estimating missing genotypes, fastPHASE was applied to the data without inferring haplotypes. Times for HaploBlock on the SeattleSNPs data and PHASE v2 on HapMap chromosome 22 are based on extrapolation from a subset of the data sets. Approximate amounts of missing data are in parentheses. The fastest method for each data set is highlighted in bold. Hap. inference = haplotype inference. .............................. 35

4.1 Error rates for missing data imputation and haplotype estimation using constant parameters of the Markov chain for cluster membership. The “unconstrained” model is the general model for which results are presented elsewhere. All results were obtained from averaging over 20 random starts of the EM. For each data set in the SeattleSNPs (50 data sets) and X-chromosome (800 data sets) data, $K$ was determined separately from among 4, 6, 8, 10 and 12. For each HapMap data set, $K$ was selected among 8, 13 and 18. ................................. 38

vii
4.2  *Sample means of \( \log q(G) \) for several distributions.* \( \log q(G) \) was calculated at 952 sites using multiple approximations \( \hat{\pi} \). Several of these are given here. The 952 values consisted of results from 6 random starts of the EM algorithm, with \( K = 12 \).  

4.3  *Genotype estimation error rates from HapMap chromosome 22.* Genotypes at 778 sites from chromosome 22 were masked at the rates of .5 (23,070 missing genotypes) and .85 (39,613 missing genotypes), and then estimated with the above methods. Results are from 5 random starts of the EM algorithm, with \( K = 5 \) and 25 iterations. “\( \theta \)-integration” averages are from 3,000 samples (plus 3,000 of equivalence class), with \( W = 10 \) and \( u = .5 \).  

5.1  *Comparison of approaches to dealing with population structure: missing data estimation and haplotype estimation.* Table entries are error rates: proportion of genotypes imputed incorrectly and switch error. Results were obtained from averaging over 20 random starts of the EM algorithm. Choice of \( K \) was made over the range of 4-12 for the SeattleSNPs data and 8-18 for HapMap data. There were 9,479 masked genotypes imputed in the SeattleSNPs data and 12,022,516 imputed for HapMap chromosome 2.  

6.1  *Distribution of SNP error types by \( \lambda \) for CEU chromosome 7.* For values of \( \lambda^* \) (natural log) from 1 to 5, the number of SNPs with \( \lambda \) values exceeding \( \lambda^* \) are given for SNPs with at least 2, at least 1, and 0 MI. The “\% of all SNPs with \( \geq 2 \) MI” column refers to the proportion detected of all multiple-MI SNPs in the collection of 60,000 SNPs from chromosome 7. FDR is calculated as the number of sites with 0 MIs divided by the number of sites with either 0 or at least 2 MIs, i.e., sites with only one MI are excluded from this quantity.  

---

viii
ACKNOWLEDGMENTS

I wish to express sincere gratitude to Matthew Stephens for the role he played in my graduate education at the University of Washington. His generosity, creativity, and critical perspectives have contributed greatly to my dissertation work and, more broadly, to my approach to research. I also wish to thank Elizabeth Thompson for her support and encouragement from the beginning of my time at UW. Her and Matthew’s comments during working group meetings have contributed a vibrant aspect to my understanding and presentation of statistical research. I was supported via two N.I.H. grants, Genome Training Grant T32 HG00035-06 and 1RO1HG/LM02585-01.

Faculty at Washington provided a rich environment for the study of statistical genetics. Ellen Wijsman and Patrick Heagerty were two of many excellent teachers; I have been fortunate to share with them discussions of research and life. I thank Bill Stewart, Bertrand Servin, Nelson Lu and Tim Robertson for their unique contributions to my education. For providing data which helped validate methods for genotype errors in this dissertation, I thank Ronald Krauss as well as Mike Eberle for assistance with intensity plots.

I wish to acknowledge the personal support I’ve received from Margo Scheet as well as Louise Strong, with whom I’ve often enjoyed chats about genetics. With Priscilla Scheet, I have had lively and helpful discussions regarding grammar and style, including edits of this dissertation; she has always been immensely supportive of all my interests and pursuits. Finally, I thank Larkin Strong, who is a perfect academic companion and continual source of love.
Chapter 1

INTRODUCTION

The human genome is organized into chromosomes, tightly packed sequences of deoxyribonucleic acid (DNA) molecules known as nucleotides or bases. Humans are a diploid species; that is, they typically have 23 pairs of chromosomes, or two copies of a (haploid) genome, inherited separately. While the genetic material of one person contains the information necessary for development and function, genetic variation among multiple individuals may elucidate an explanation for variation of disease susceptibility or differential response to environmental forces.

The ultimate source of genetic variation in a population is some form of DNA mutation. Point mutations, the change of single nucleotides, may propagate through the population over time to a detectable frequency, either by selective advantage (conferment of above-average reproductive success) or genetic drift (repeated sampling from a finite population). Such a variant is called a single nucleotide polymorphism (SNP) and is a relatively common form of genetic variation, occurring on the order of once per every few hundred to one thousand bases between 2 haploid genomes. Genetic markers denote sites in the genome, or loci, where chromosomes exhibit variation. Variants at marker loci are called alleles; typically, there are two alleles for SNPs.

The principal aim of this thesis is to develop models for genetic variation in large random samples of unrelated individuals taken from a population. Figure 1.1 illustrates the kinds of data and patterns I wish to model. A key feature of these kinds of data is a (occasionally very strong) correlation among nearby sites on a chromosome.
These correlations are referred to as linkage disequilibrium (LD) and can occur via several mechanisms. Next, I give a more detailed introduction to LD.

1.1 Linkage Disequilibrium

A new mutant allele is immediately associated with the genetic background, or allelic configuration at nearby genetic markers, of the chromosome on which the mutation occurred. This association tends to persist through many generations due to linkage, the cosegregation of alleles on the same chromosome during the formation of gametes. Correlation of allelic types, or LD, at multiple loci may also result from other forces, such as the existence of population structure, e.g., admixture (recent mixing of two previously-distinct populations) and cryptic subpopulations (non-random mating).

Whereas mutation acts to create LD between two loci, recombination breaks it down by allowing a mutant allele to be on the same chromosome as various alleles at other loci. Historical mutation and recombination throughout the genealogy of a set of chromosomes help shape the patterns of human genetic variation. The sequence of alleles on a chromosomal segment form a haplotype, and a specific diploid configuration of allelic variants for an individual is a genotype. Since haplotypes represent coinherited allelic variants, it is natural to model genetic variation by developing models for haplotypes rather than directly for the genotypes.

Before I turn to a discussion of previous approaches to modelling LD among multiple loci, I briefly review previous efforts at measuring and describing patterns of LD, beginning with the two-locus case.

1.1.1 Measures of LD

Let $p_{AB}$ denote the frequency of chromosomes carrying the $A$ and $B$ alleles at 2 polymorphic loci, and let $p_A$ denote the frequency of the $A$ allele in a population. Consider two polymorphic loci with alleles $A$ and $a$ at the first locus and $B$ and $b$ at the second. Robbins (1918) introduced a measure for allelic association between
Figure 1.1: *Dense genotype data.* Genotypes from 23 African Americans and 24 CEPH individuals in the SeattleSNPs database are represented visually. Each individual’s genotypes make up a row of the plot, with columns consisting of all the sampled genotypes at a SNP. Every colored square denotes a genotype: blue for the genotype consisting of two copies of the most common allele at a SNP (column); yellow for 2 copies of the less-common allele; red for one copy of each; and gray for missing genotypes. The image was created with the *Visual Genotypes* software on the SeattleSNPs website.
two loci, defined as $\Delta = p_{AB}p_{ab} - p_{Ab}p_{aB}$. This is algebraically equivalent to the commonly-used measure for LD, $D = p_{AB} - p_{APB}$, given by Lewontin & Kojima (1960).

Two other measures of LD common in the literature are $D'$ (Lewontin, 1964), a normalized version of $D$ which has $(-1, 1)$ as its range, and $\delta = D/(p_{APa}p_{Bp_b})$ (Hill & Robertson, 1968), known as a coefficient of correlations.

These measures describe the pairwise LD that exists between two markers; of course, allelic dependence exists among multiple loci, as well. Maniatis et al. (2002) introduce an approach for examining multilocus LD that involves the calculation and combination of pairwise statistics among all polymorphic loci.

Finally, a common approach is to represent the pairwise statistics in a visual fashion, such as in square plots (for examples, see Clark et al., 1998; Jeffreys et al., 2001).

1.1.2 Haplotype "Blocks"

Pairwise measures, while easy to interpret, do not naturally capture the multilocus dependencies that exist in actual population data. It has been observed that for certain regions of the genome, haplotype diversity is limited, with relatively few haplotypes representing a large fraction of the total haplotypes observed in a sample (Patil et al., 2001; Daly et al., 2001). Such regions are often referred to as blocks.

Wall & Pritchard (2003) classify definitions for haplotype blocks into two groups: those that define blocks as regions of limited haplotype diversity, and those that use a definition based on regions of high pairwise LD among consecutive markers. Even with formal definitions, there are numerous ways to divide genomic regions into blocks, and the task of partitioning blocks presents a computational problem.

The concept of blocks was stimulated by empirical observation in sampled haplotypes; they are less relevant if they are not representative of future-sampled haplotypes. Recombination and mutation events that occur early in a genealogy (of
haplotypes) may dominate the population structure observed many generations later, leading to apparent blocks (Clark, 2004).

1.1.3 Modelling Haplotype Variation

Descriptive approaches, such as 2-locus measurements for LD and haplotype blocks, do not provide an explicit model; therefore, they are limited in terms of the types of inference they might provide. For example, it is unclear how to utilize such measures to predict the genotypes at sites where observations are missing, or to make probabilistic statements regarding haplotype variation, as one might do when reconstructing haplotypes from unphased genotype data. Generally, formal inference will require a probability model. This provides a more comprehensive description of the dependence among multiple alleles. Additionally, it allows for more direct inference.

One stochastic model for variation in a collection of homologous DNA sequences is the coalescent (Kingman, 1982), which was extended by Hudson (1983) to account for recombination. A fundamental consequence of the coalescent model for DNA sequences is that they derive from a common ancestor.

Two compound parameters of interest in a coalescent model are \( \rho = 4Nc \) and \( \Theta = 4N\mu \), where \( c \) and \( \mu \) are the per basepair (per meiosis) recombination and mutation rates, respectively, and \( N \) is the effective population size (see Ewens, 1979). Methods have been proposed for their estimation from population data (Wilson & Balding, 1998; Kuhner et al., 2000; Fearnhead & Donnelly, 2001; Li & Stephens, 2003). Methods which could accurately incorporate all features of the coalescent would be computationally intractable and certain approximations must be made.

Composite Likelihood

Hudson (2001) introduced a composite-likelihood method for estimating \( \rho \) based on the sampling distribution of 2-loci gamete counts. The probability model for one such
pairwise statistic involves \( \rho \), and the likelihood for \( k \) pairs of polymorphic sites is approximated as the product over all \( k \) pairs of the marginal probability mass function for each pair. McVean et al. (2002) extended this model to handle repeat mutation, and Fearnhead & Donnelly (2002) used an approximate (composite) likelihood technique by ignoring long-range dependencies.

These methods have the advantage of being computationally tractable while still incorporating features of the coalescent process; however, since they do not consider all of the data concurrently, a loss of power may result.

**PAC Model**

The method of a product of approximate conditionals (PAC) was proposed by Li & Stephens (2003). It relates the distribution of a collection of haplotypes to the recombination parameter \( \rho \) via a product of conditional probabilities. These conditional probabilities are chosen in such a way that they are computationally tractable, capturing the key qualitative features that one might expect to see under a coalescent model. Suppose we sample \( n \) haplotypes, \( \mathbf{h} = (h_1, \ldots, h_n) \). The probability of \( \mathbf{h} \) is a function of \( \rho \) and approximated as:

\[
\Pr(h_1, \ldots, h_n; \rho) \approx \hat{\pi}(h_1; \rho)\hat{\pi}(h_2|h_1; \rho)\cdots\hat{\pi}(h_n|h_1, \ldots, h_{n-1}; \rho).
\]  

(1.1)

The quality of the approximation depends on the accuracy of the estimated conditional distribution, \( \hat{\pi} \). Li & Stephens (2003) review several known approximations for the true conditional distribution, \( \Pr(h|h^*; \rho) \), under the assumption that the observed sequences arose under a coalescent model. Stephens & Donnelly (2000) proposed an approximation based upon the coalescent which did not explicitly model recombination but which provided relatively accurate models for haplotypes from a coalescent process with relatively low recombination rates, evidenced by success of a haplotype reconstruction method based on this approximation (Stephens et al., 2001). Fearnhead & Donnelly (2001) extended this result to produce a form for \( \hat{\pi} \) which accounted...
for recombination. Li & Stephens (2003) used a simplification of this extension for
their computation of a PAC likelihood, $L_{PAC}$ which equals the right-hand side of
(1.1).

The basic principle, from Stephens & Donnelly (2000), is that a sampled haplotype
will tend to look similar to an existing haplotype; that is, it may look identical,
but if it varies, it will vary at relatively few positions. This heuristic mimics the
expected distribution of a collection of haplotypes derived from the coalescent without
recombination. Stephens & Donnelly (2000) assume a geometric distribution for the
number of mutations between a newly sampled haplotype and a previously-sampled
capture the concept that a sampled haplotype will look like an imperfect mosaic of
previously-sampled haplotypes, due to recombination events between the ancestors
of sampled chromosomes. Li & Stephens (2003) accomplish this by using a Markov
chain along the sequence of the haplotype to model the source haplotype from which
copying occurs.

Li & Stephens (2003) apply their PAC methods to estimate $\rho$. However, as a model
for haplotype variation, further applications are possible. Since the computation of
(1.1) involves approximations to the true (unknown) conditional distributions, the
quantity is dependent on the order in which the sampled haplotypes are considered.
In theory, this could be overcome by considering every $n!$ ordering; their solution is
to average over random permutations of orderings. The computational complexity of
the calculation of $L_{PAC}$ is $O(n^2M)$, where $n$ is the number of sampled chromosomes
and $M$ is the number of typed markers. Since the complexity grows with the square
of the sample size, this method may be rather intensive for large $n$. Further, it cannot
be applied directly to genotype data. Direct applications to genotypes requires a
time-consuming and difficult-to-implement MCMC scheme to estimate the haplotype
reconstructions and recombination rates jointly (Crawford et al., 2004; Stephens &
Scheet, 2005).
1.2 Motivation for a Flexible and Computationally Tractable Model

The coalescent captures the following features of observed haplotype data: many haplotypes tend to look similar to each other, varying at a few sites; and some haplotypes vary considerably in their sequence of allelic patterns. The observed similarity is due to the common ancestry of a collection of chromosomes. Many of these realistic features, including the expectation that, due to recombination, LD decays with increasing distance, are mimicked by the PAC model. However, the PAC model is computationally intractable for samples of thousands of individuals typed at thousands of polymorphic loci.

Here I present an efficient and accurate alternative to the PAC model for modelling multilocus haplotype or genotype data, feasible for tens of thousands of individuals and hundreds of thousands of polymorphic sites per chromosome. In chapter 2, I present a model for unphased multilocus genotypes, which is based on clustering of haplotypes into groups of similar haplotypes and allowing the cluster memberships to change along the chromosome in a continuous fashion.

Subsequent chapters contain applications and extensions of the model. I present methods for estimating missing genotypes and inferring haplotypes based on this model in Chapter 3, comparing the results from its application to real data with results obtained from multiple state-of-the-art methods for these tasks. Chapter 4 contains several extensions to the models presented in Chapter 2, including the following: 1) the imposition of constraints during parameter estimation, and 2) a technique that utilizes information at typed loci to approximate the distribution of genotypes at untyped loci. I present a model which takes account of certain deviations from the situation of completely randomly-mating populations in Chapter 5. In Chapter 6, I extend the model to account explicitly for genotyping error and consider an application to error detection in population genetic data. Finally, Chapter 7 contains a review of the contributions of this thesis and a brief discussion of future work.
Chapter 2

A MIXTURE MODEL FOR HAPLOIDS AND DIPLOIDS

To introduce notation and the basic concepts underlying the model, I begin by describing a simple cluster-based model for haplotypes sampled from a population, in which each haplotype is assumed to have arisen from a single cluster. I then describe a modification to this model that allows cluster membership to change along each haplotype, to capture the fact that, while sampled haplotypes exhibit cluster-like patterns, these patterns tend to be local in nature. Finally, I describe the extension of this model for haplotype data to unphased genotype data through the assumption of Hardy–Weinberg equilibrium (HWE). Parameter estimation is accomplished via an Expectation–Maximization algorithm with details given in an appendix.

2.1 A Cluster Model for Haplotypes

Consider a sample of $n$ haplotypes, $h = (h_1, \ldots, h_n)$, each comprised of data at $M$ markers. Let $h_{im}$ denote the allele in the $i^{th}$ haplotype at marker $m$, so $h_i = (h_{i1}, \ldots, h_{iM})$. Throughout, I assume the markers are biallelic SNPs, with alleles labeled 0 and 1 (arbitrarily) at each site, although the model may be extended to allow for multi-allelic markers.

A simple cluster model for haplotypes can be developed as follows. Assume that each sampled haplotype originates from one of $K$ clusters (labeled $1, \ldots, K$). For simplicity, I assume initially that $K$ is known but will relax this assumption later. Let $z_i$ denote the (unknown) cluster of origin for $h_i$, and let $\alpha_k$ denote the relative frequency of cluster $k$, so that $p(z_i = k | \alpha) = \alpha_k$, where $\alpha = (\alpha_1, \ldots, \alpha_K)$.

I assume that, given the cluster of origin of each haplotype, alleles observed at
each marker are independent draws from cluster-specific (and marker-specific) allele frequencies. Thus if \( \theta_{km} \) denotes the frequency of allele 1 in cluster \( k \) at marker \( m \), and \( \theta \) denotes the matrix of these values, \( h_{i1}, \ldots, h_{iM} \) are conditionally independent and

\[
p(h_i | z_i = k, \theta) = \prod_{m=1}^{M} \theta_{km} h_{im} (1 - \theta_{km})^{1-h_{im}}.
\]

Since the clusters of origin are actually unknown, the probability of \( h_i \) is obtained by summing (2.1) over all possible values for \( z_i \), weighting by their probabilities:

\[
p(h_i | \alpha, \theta) = \sum_{k=1}^{K} p(z_i = k | \alpha) p(h_i | z_i = k, \theta)
= \sum_{k=1}^{K} \alpha_k \prod_{m=1}^{M} \theta_{km} h_{im} (1 - \theta_{km})^{1-h_{im}}.
\]

Finally, specification of the model for \( h = (h_1, \ldots, h_n) \) is completed by assuming that \( h_1, \ldots, h_n \) are independent and identically distributed (\textit{iid}) from (2.2).

This simple model is essentially a haploid version of a model that has been widely used to capture the clustering that can occur among individuals typed at unlinked (or loosely-linked) markers due to population structure in natural populations (Smouse et al., 1990; Rannala & Mountain, 1997; Pritchard et al., 2000). However, in those applications the clusters represent “populations”, whereas here the clusters represent groups of closely-related haplotypes. The idea of using this model to capture clustering of haplotypes at tightly-linked markers seems to originate with Koivisto et al. (2003), who used it to model data at tightly-linked SNPs within a “haplotype block” (see also Kimmel & Shamir, 2005).

Each cluster corresponds to a single row of the \( \theta \) matrix, which is a vector of numbers in the range \([0, 1]\), one number per marker. (In fact, I imposed a constraint \( 0.001 \leq \theta_{km} \leq 0.999 \) on the elements of \( \theta \), motivated by the idea that this might make the model more robust to factors such as genotyping error.) Ideally, each row of \( \theta \) will look like a haplotype (a string of 0s and 1s), but with occasional “fuzziness” indicating uncertainty about the alleles at some positions.
2.2 Local Clustering of Haplotypes

Although sampled haplotypes certainly exhibit cluster-like patterns, these patterns tend to be local in nature (Figure 2.1). To capture this I replace the assumption that each haplotype originates from one of $k$ clusters with the assumption that each allele originates from one of the clusters, using a hidden Markov model (HMM) to model the fact that alleles at nearby markers are likely to arise from the same cluster. Specifically, if $z_{im}$ denotes the cluster of origin for $h_{im}$, I assume $z_i = (z_{i1}, \ldots, z_{iM})$ forms a Markov chain on $\{1, \ldots, K\}$, with initial state probabilities

$$p(z_{i1} = k) = \alpha_{k1},$$  \hspace{1cm} (2.3)

and locus-specific transition probabilities $p_m(k \rightarrow k')$ given by:

$$p_m(k \rightarrow k') := p(z_{im} = k' | z_{i(m-1)} = k, \alpha, r) = \begin{cases} 
  e^{-r_m d_m} + (1 - e^{-r_m d_m}) \alpha_{k'i}, & k' = k, \\
  (1 - e^{-r_m d_m}) \alpha_{k'i}, & k' \neq k,
\end{cases}$$  \hspace{1cm} (2.4)

for $m = 2, \ldots, M$, where $d_m$ is the physical distance between markers $m-1$ and $m$ (assumed known), and $r = (r_2, \ldots, r_M)$ and $\alpha = (\alpha_{km})$ are parameters to be estimated. This Markov chain is a discretized version of a continuous Markov jump process, with jump rate $r_m$ per basepair between markers $m-1$ and $m$, and transition probabilities

$$p(z_{im} = k' | z_{i(m-1)} = k, \text{ jump occurs}) = \alpha_{k'i},$$  \hspace{1cm} (2.5)

Informally, $r_m$ is related to the recombination rate between $m-1$ and $m$, although there may generally be little correspondence between actual recombination rate and estimates of $r$. If the physical distances between markers were not known, then the compound parameter $r_m d_m$ in (2.4) could be replaced by a single parameter without loss of information. Indeed, this is true even if the physical distances are known, unless some constraint is placed on $r$ (e.g., constraining all $r_m$ to be equal).
Given the cluster of origin of each allele, the alleles are independent draws from the relevant cluster allele frequencies, so

\[
p(h_i | z_i, \theta) = \prod_{m=1}^{M} p(h_{im} | z_{im}, \theta), \tag{2.6}
\]

where

\[
p(h_{im} | z_{im} = k, \theta) = \theta_{km}^{h_{im}} (1 - \theta_{km})^{1 - h_{im}}. \tag{2.7}
\]

Since \( z_i \) is unknown, the probability of \( h_i \) is obtained by summing (2.6) over all possible values for \( z_i \), weighting by their probabilities:

\[
p(h_i | \alpha, \theta, r) = \sum_{z_i} p(z_i | \alpha, r) p(h_i | z_i, \theta), \tag{2.8}
\]

where \( p(z_i | \alpha, r) \) is determined by (2.3) and (2.4). Naïve computation of this sum would require a sum over \( K^M \) possible values for \( z_i \), but the Markov assumption on \( z_i \) allows the sum to be computed much more efficiently (computational cost increasing linearly with \( K M \)) using standard methods for HMMs (e.g. Ewens & Grant, 2001).

### 2.3 Extension to Genotype Data

Suppose now that instead of observing haplotypes, I observe unphased genotype data \( g = (g_1, \ldots, g_n) \) on \( n \) diploid individuals. Let \( g_{im} \) denote the genotype at marker \( m \) in individual \( i \), which I will code as the sum of its alleles, so \( g_{im} \) takes values 0, 1 or 2. One approach to extending the haplotype-based model above to unphased genotype data is to assume that the two haplotypes that make up each multilocus genotype are \( iid \) from (2.8), i.e. to assume HWE. Under this assumption, if \( z_{im}^* \) denotes the (unordered) pair of clusters from which genotype \( g_{im} \) originates, then \( z_i^* = (z_{i1}^*, \ldots, z_{iM}^*) \) forms a Markov chain with initial state probabilities

\[
p(z_{i1}^* = \{k_1, k_2\}) = \begin{cases} (\alpha_{k_1})^2, & k_1 = k_2, \\ 2\alpha_{k_1}\alpha_{k_2}, & k_1 \neq k_2, \end{cases} \tag{2.9}
\]
Figure 2.1: Illustration of how the model allows cluster membership to change continuously along a chromosome. Each column represents a SNP, with the two alleles indicated by open and crossed squares. Successive pairs of rows represent the estimated pair of haplotypes for successive individuals. Colors represent estimated cluster membership of each allele, which changes as one moves along each haplotype. Locally each cluster can be thought of as representing a (common) combination of alleles at tightly-linked SNPs, and the figure illustrates how each haplotype is modelled as a mosaic of these common combinations. The figure was produced by fitting the model to the HapMap data from 60 unrelated CEPH individuals (Section 3.4.1), and then taking a single sample of cluster memberships and haplotypes from their conditional distribution, given the genotype data and parameter estimates (Appendix B). For brevity, haplotypes from only 10 individuals are shown.
and transition probabilities

\[
p_m(\{k_1, k_2\} \rightarrow \{k'_1, k'_2\}) = \begin{cases} 
  p_m(k_1 \rightarrow k'_1)p_m(k_2 \rightarrow k'_2) + \\
  p_m(k_1 \rightarrow k'_2)p_m(k_2 \rightarrow k'_1), & k_1 \neq k_2 \text{ and } k'_1 \neq k'_2, \\
  p_m(k_1 \rightarrow k'_1)p_m(k_2 \rightarrow k'_2), & \text{otherwise};
\end{cases}
\]

(2.10)

where \(p_m(k \rightarrow k')\) is defined in (2.4). These expressions come from pairing two independent Markov chains with transition probabilities (2.4).

Given the clusters of origin, \(z^*_i\), I again assume that the alleles are independent draws from the relevant cluster allele frequencies, so

\[
p(g_i | z^*_i, \theta) = \prod_{m=1}^{M} p(g_{im} | z^*_m, \theta),
\]

(2.11)

where

\[
p(g_{im} | z^*_m = \{k_1, k_2\}, \theta) = \begin{cases} 
  (1 - \theta_{k1m})(1 - \theta_{k2m}), & g_{im} = 0, \\
  \theta_{k1m}(1 - \theta_{k2m}) + \theta_{k2m}(1 - \theta_{k1m}), & g_{im} = 1, \\
  \theta_{k1m}\theta_{k2m}, & g_{im} = 2.
\end{cases}
\]

(2.12)

Note that if some \(g_{im}\) are missing, this is easily dealt with by replacing the corresponding \(p(g_{im} | z^*_m = \{k_1, k_2\}, \theta)\) with any positive constant (e.g. 1.0); this corresponds to an assumption that the genotypes are missing at random.

Since \(z^*_i\) is unknown, the probability of \(g_i\) is obtained by summing over all possible values:

\[
p(g_i | \alpha, \theta, r) = \sum_{z^*_i} p(z^*_i | \alpha, r)p(g_i | z^*_i, \theta),
\]

(2.13)

where \(p(z^*_i | \alpha, r)\) is determined by (2.9) and (2.10). As before, methods for HMMs allow this sum to be computed efficiently (computational cost increasing linearly with \(K^2M\)); see Appendix A.

This model (2.13) is reminiscent of the “linkage” model of Falush et al. (2003), who modelled genotype data at loosely-linked markers in structured populations. One
difference between their model and this one is that they allowed $\alpha$ ($g$ in their notation) to vary among individuals, but fixed $\alpha$ across markers, whereas I allow $\alpha$ to vary across markers but assume it to be fixed across individuals. The reason for this difference is that the interpretation of these parameters is very different in the two applications; in Falush et al. (2003) this parameter controls each individual's proportion of ancestry in each subpopulation (which would be expected to differ across individuals), whereas here it controls the relative frequency of the common haplotypes (which would be expected to differ in different genomic regions). Falush et al. (2003) also restricted $r$ to be constant, whereas I allow it to vary in each marker interval.

### 2.4 Parameter Estimation

In this section I outline the methods used to fit the model for diploid data (2.13). I use an Expectation–Maximization (EM) algorithm (Dempster et al., 1977) to estimate the parameters, $\nu = (\theta, \alpha, r)$, of the models presented here (see Appendix C for details). The computational complexity of the algorithm for parameter estimation and calculation of the likelihood (2.13) is $O(nMK^2)$, and in particular is linear in the number of sampled individuals and markers, which allows it to be fitted to large datasets.

As with any EM algorithm, this algorithm will typically find a local maximum of the likelihood function, $\mathcal{L}(\nu; g) = p(g|\nu)$. For realistic datasets this likelihood surface will have many different local maxima, and so different starting points for the EM algorithm will typically lead to different parameter estimates. A standard approach to dealing with this problem is first to apply the algorithm $T$ times, from $T$ different starting points, to obtain $T$ estimates $\hat{\nu}_1, \ldots, \hat{\nu}_T$, and then to select whichever of these estimates gives the highest value for the likelihood. However, because the focus here is on using the model for prediction, and not on parameter estimation itself, it is not necessary to settle on a single estimate for the parameters, and in tests I found that I was able to obtain more accurate predictions by combining results across $T$
estimates, as described below. I also found that, using this strategy of combining across estimates, reliable performance can be obtained with relatively few iterations of the EM algorithm per starting point — probably far fewer than would be required by most methods of monitoring convergence.

For the results presented later I typically used $T = 20$ starts of the EM algorithm, with up to 25 iterations per start, although experiments (results not shown) suggest that these values could be reduced without sacrificing accuracy. For each initialization of the EM algorithm, I set $1 - e^{-m\cdot\theta_{km}} = .01$, chose $\theta_{km}$ to be i.i.d. uniform on [.001, .999], and $\alpha_{km} = \frac{1}{K}$, for all $m$ and $k$.

As a confirmation of correct implementation of the EM algorithm, I note that in all situations encountered, the likelihood strictly increases during the EM algorithm (for the model presented here, as well as extensions in subsequent chapters). Additionally, calculation of the likelihood via the forward and backward algorithms (Appendix A) yields equivalent results.

2.4.1 Selecting $K$

Selecting $K$ is essentially a “model selection” problem, which is, in general, a tricky statistical problem. I found that standard approaches to model selection, such as $AIC$ Akaike (1973) and $BIC$ Schwarz (1978), do not work well here (selecting $K$ too small), probably because the asymptotics on which they are based do not apply. I therefore used the following cross-validation approach to select $K$. For each dataset I masked (i.e., made missing) approximately 15% of the genotypes at random. Then, for a range of values of $K$ (I considered $K = 4, 6, 8, 10, 12$, the upper limit reflecting a desire to keep the computational burden down), I estimated the masked genotypes (as described later), comparing these estimates with the true genotypes. I selected the $K$ that maximized the number of correctly-estimated genotypes.

This procedure is relatively computationally intensive, as it involves fitting the model for several values of $K$, and the computational cost increases with $K^2$. To
reduce the computation burden, when performing the cross-validation I used only a small number of starts (as few as 3) for the EM algorithm. In addition, I sometimes chose a single $K$ for several datasets on a common set of individuals, or applied cross-validation to only a portion of the data for larger data sets.

Since in general I found performance to be relatively robust to a range of values of $K$, in cases where the computational cost of this strategy becomes inconvenient, an alternative approach would be to simply select a fixed value of $K$. It would also be possible to combine results across parameter estimates obtained using different values of $K$, rather than selecting a single $K$.

2.5 Remarks

This model is the foundation for all that follows in subsequent chapters, including alternatives to parameter estimation of Appendix C in Chapter 4 and the introduction of a slightly more general parameterization of the model in Chapter 5. Chapter 6 contains an extension to the stochastic model presented here.

The model, as presented in this chapter, allows for probability statements regarding unobserved data. For example, if a single genotype is missing (for one individual at one SNP), I can use the model to calculate probabilities for its possible values. Likewise, this can be done for unobserved haplotypes, given unphased genotype data. These applications are presented in the next chapter.
Chapter 3

INFERENCE OF MISSING GENOTYPES AND HAPLOTYPES

In this chapter I evaluate the performance of the model presented in Chapter 2 in terms of its ability to estimate two quantities from unphased SNP genotype data: 1) unobserved alleles in the form of missing unphased genotypes and 2) haplotypes, given the unphased genotypes. For comparison, I also evaluate the performance of multiple existing methods for these tasks.

3.1 Review of Existing Methods

The first method to estimate haplotypes from unphased genotype data was a greedy algorithm proposed by Clark (1990). Clark’s method first obtains the unambiguous haplotypes from samples which have at most one heterozygous locus. These “template” haplotypes form a list from which complementary haplotypes in ambiguous samples may be inferred, with the new (complementary) haplotypes added to the list of known haplotypes. This method has several drawbacks, including the requirement of unambiguous haplotypes in the sample for initiation of the algorithm. Subsequently several authors proposed a multinomial model for haplotypes, using an EM algorithm for frequency estimation (Excoffier & Slatkin, 1995; Hawley & Kidd, 1995). These methods, along with Clark’s algorithm, are compared in Stephens et al. (2001).

The methods I consider here are summarized in Table 3.1. Several of the methods assume a multinomial model for haplotypes (Bayesian Dirichlet, HaploTyper, PL–EM, snphap); however, computational tricks and approximations were introduced which allow for application to larger data sets than were computationally feasible based
on the original (1995) implementations. For example, Niu et al. (2002) introduced “partition ligation” (PL) as a computational trick to handle larger numbers of SNP markers per data set. It works by first dividing data sets consisting of many markers into segments containing a small number of contiguous markers and then inferring haplotypes within each segment before iteratively combining results from adjacent segments. This framework was subsequently adopted by others, including Qin et al. (2002), Stephens & Donnelly (2003) and Lin et al. (2004). More-complete reviews may be found in the cited references of Table 3.1.

The methods presented in Table 3.1 were proposed for the reconstruction of haplotypes. However, they also make an attempt to impute genotypes where data are missing, as well, based on the models underlying the methods for haplotype inference.

3.2 Missing Genotype Imputation

For any genotype \( g_{im} \) that is unobserved (“missing”) it is straightforward to compute the probability that \( g_{im} = x \) \( (x = 0, 1, 2) \), given all observed genotypes \( g \) and parameter values \( \nu \), using

\[
p(g_{im} = x | g, \nu) = \sum_{k_1=1}^{K} \sum_{k_2=k_1}^{K} p(g_{im} = x | z_{im} = \{k_1, k_2\}, \nu) p(z_{im} = \{k_1, k_2\} | g, \nu).
\]

(3.1)

The first factor in this sum is given by (2.12), and the second is the conditional distribution of the hidden variables in the HMM, which can be obtained using standard methods for HMMs (Appendix A).

A natural point estimate for \( g_{im} \) is then obtained by choosing the value of \( x \) that maximizes this expression. As noted above, I have found it helpful to combine results over several sets of parameter estimates \( \hat{\nu}_1, \ldots, \hat{\nu}_T \), obtained from \( T \) different applications of the EM algorithm using different starting points. Specifically, I used the estimate

\[
\hat{g}_{im} = \arg \max_{x \in \{0,1,2\}} \frac{1}{T} \sum_{t=1}^{T} p(g_{im} = x | g, \hat{\nu}_t).
\]

(3.2)
Table 3.1: *Summary of haplotype inference algorithms for unrelated individuals.*

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHASE v2</td>
<td>Bayesian method with approximate “coalescent with recombination” prior, capturing fact that each sampled haplotypes will tend to be similar to another haplotype, or to a mosaic of other haplotypes.</td>
<td>Stephens &amp; Scheet (2005)</td>
</tr>
<tr>
<td>(default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>−MQ hybrid</td>
<td>Hybrid algorithm, reduces computing time over MR by assuming no recombination for the majority of the computation, before incorporating recombination for the final steps.</td>
<td>Stephens &amp; Scheet (2005)</td>
</tr>
<tr>
<td>−MS no recom</td>
<td>Bayesian method with approximate “coalescent without recombination” prior, based on the idea that sampled haplotypes look similar to other haplotypes. Ignores decay of LD with distance.</td>
<td>Stephens et al. (2001), Stephens &amp; Donnelly (2003)</td>
</tr>
<tr>
<td>GERBIL</td>
<td>Algorithm partitions loci into blocks of low haplotype diversity; within a block, haplotypes assumed to come from distinct clusters (cf. HaploBlock, HAP). Parameter estimation via EM algorithm.</td>
<td>Kimmel &amp; Shamir (2005)</td>
</tr>
<tr>
<td>HaploBlock</td>
<td>Loci partitioned into blocks; within blocks, haplotypes assumed to come from a mixture model (cf. GERBIL). Predicted haplotypes combined over multiple points in the “parameter” space.</td>
<td>Greenspan &amp; Geiger (2004)</td>
</tr>
<tr>
<td>v1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayes Dirichlet</td>
<td>Bayesian method, with Dirichlet prior on population haplotype frequencies; ignores similarity of haplotypes. (Implementation of the algorithm in Haplotyper, below.)</td>
<td>Stephens et al. (2001), Niu et al. (2002)</td>
</tr>
<tr>
<td>Haplotyper</td>
<td>Bayesian method with a Dirichlet prior on the haplotype frequencies; ignores similarity of haplotypes.</td>
<td>Niu et al. (2002)</td>
</tr>
<tr>
<td>PL-EM</td>
<td>A maximum likelihood approach; uses EM algorithm to obtain estimates of haplotype frequencies, and uses these estimates to estimate most probable haplotypes for each individual.</td>
<td>Qin et al. (2002)</td>
</tr>
<tr>
<td>hap2</td>
<td>Based on a Dirichlet prior for population haplotype frequencies but looks for matches only at heterozygous sites; in this sense it does not base inference only on “exact” matches, and so could be said to take some account of similarity in an ad hoc way.</td>
<td>Lin et al. (2004)</td>
</tr>
<tr>
<td>HAP</td>
<td>First partitions loci into blocks; assumes that haplotypes within blocks will conform approximately to a “perfect phylogeny” (i.e. no recombination or repeat mutation) (cf. GERBIL, HaploBlock).</td>
<td>Eskin et al. (2003), Halperin &amp; Eskin (2004)</td>
</tr>
<tr>
<td>Arlequin 3.0a</td>
<td>Bayesian method. Prior takes some account of similarity of haplotypes. Bases inference for each locus on data in a window of nearby loci, which is allowed to vary in size based on local levels of LD.</td>
<td>Excoffier et al. (2003)</td>
</tr>
<tr>
<td>ELB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This method imputes genotypes marginally and provides a “best guess” for each genotype. It is also straightforward to sample from the joint distribution of the missing genotypes given observed data, for example by sampling from the conditional distribution of the haplotypes for all individuals as described below.

3.3 Haplotype Estimation

I consider two aspects of the haplotype inference problem: 1) Sampling the pairs of haplotypes of all individuals, from their joint distribution given the unphased genotype data. This provides a useful way to assess or account for uncertainty in haplotype estimates. 2) Constructing point estimates of the haplotypes carried by each individual. This is how many haplotype reconstruction methods are used in practice (as a prelude to subsequent analysis of the estimated haplotypes). It also provides a convenient basis for comparison with other haplotype reconstruction methods.

Sampling haplotypes from their joint distribution I will use diplotype to refer to a pair of haplotypes that comprise the genetic data for an individual. Let $d_i$ denote the diplotype for individual $i$, and $d = (d_1, \ldots, d_n)$. Conditional on a particular parameter value $\nu$, the diplotypes of different individuals are independent ($p(d|g, \nu) = \prod_i p(d_i|g_i, \nu)$), and so one can sample from $p(d|g, \nu)$ by sampling independently from $p(d_i|g_i, \nu)$ for each $i$. A method for doing this is described in Appendix B.

To combine results from $T$ initiations of the EM algorithm, I obtain a sample $\tilde{d}$ of size $T \times B$ as the union of $B$ independent samples from each of $p(d|g, \tilde{\nu}_1), \ldots, p(d|g, \tilde{\nu}_T)$. When constructing $\tilde{d}_i^{SW}$, I used $T = 20$ and $B = 50$; for $\tilde{d}_i^{SN}$ I used $T = 20$ and $B = 200$.

Point estimation I have implemented two different methods for producing point estimates, $\hat{d}_i$, of the diplotypes of individual $i$. Both are obtained by first creating a sample $\hat{d}_i$ of diplotypes for individual $i$, as described above. From this sample I define:
1. \( \hat{d}_i^{RN} \) to be the diploptype that appears most often in \( \tilde{d}_i \). This estimate is motivated by attempting to maximize the probability that the whole diploptype is correct, or equivalently to minimize the "individual error rate" (Stephens et al., 2001), being the proportion of ambiguous individuals whose estimated diplotypes are not entirely correct.

2. \( \hat{d}_i^{SW} \), which is constructed as follows. Starting at one end of the genetic region, I move through the heterozygous sites, phasing each relative to the previous heterozygous site by selecting the 2-site diploptype that occurs most frequently (at that pair of sites) in \( \tilde{d}_i \). This estimate is motivated by attempting to minimize the "switch error", being the proportion of heterozygous sites that are phased incorrectly relative to the previous heterozygous site (this is 1 minus the switch accuracy of Lin et al., 2002).

Note that for datasets containing a large number of markers, individuals may have a very large number of plausible diploptype configurations, none of which is overwhelmingly more probable than the others. In such cases the most probable diploptype configuration is both difficult to identify reliably (requiring a very large sample \( \tilde{d}_i \)) and not especially interesting (in that it is probably not correct). Therefore, for large-scale studies, I favor the use of \( \hat{d}_i^{SW} \) over \( \hat{d}_i^{RN} \).

### 3.4 Results

The methods described above in Sections 3.2 and 3.3 for imputing missing genotype data and estimating haplotypes are implemented in a software package called fastPHASE. Here I compare performance of these methods with those in Table 3.1.

The models underlying both GERBIL and HaploBlock bear some similarity to my model, being based on the idea of clusters of haplotypes, but with cluster membership being allowed to change only at certain points in the genome ("block-boundaries"), which are estimated from the data.
The model underlying PHASE is based on the PAC model of Li & Stephens (2003), which shares the flexibility of the model for haplotypes of Section 2.8, but is considerably more costly to compute. For haplotype estimation, Kimmel & Shamir (2005) found that GERBIL performed better than HaploBlock, but slightly less well than PHASE. However, they did not examine accuracy for missing data imputation.

I ran all computer programs except Arlequin 3.0a with their default or recommended settings, except as noted below. (Results for Arlequin 3.0a were supplied by L. Excoffier.) For HaploTyper and PL-EM, “rounds” was set to 20. HAP was accessed via the HAP Webserver (accessed October 2003 and December 2003). For snphap, I reduced the lower posterior-trimming threshold to 0.001, in an attempt to obtain haplotype reconstructions for all individuals in the sample. The input format for hap2 (provided by S. Lin) allows for the inclusion of family-level data; I input the individuals as unrelated parents with no offspring, and I used the following additional settings: 10,000 iterations, 5,000 burn-in, 20 thinning, (0, 0.5) minor-allele frequency limits, and \(|D|\) blocks with 0.8 threshold. HaploBlock was run with the -W option (which produces estimates by combining over multiple solutions, similar to my strategy of combining estimates from multiple runs of the EM algorithm; this seemed to give more accurate results, but took longer to run, than the alternative -F option).

3.4.1 Missing Genotype Imputation

I examined accuracy of methods for missing data imputation on both complete sequence data in multiple candidate genes (from the SeattleSNPs Variation Discovery Resource) and for genome-wide dense SNP data (1 SNP every 2-3 kilobases across entire chromosomes) from The International HapMap Project (International HapMap Consortium, 2005).

To provide a baseline against which to compare methods, I implemented a naive “straw man” approach to imputing the genotypes. This method imputes the most common genotype in the sample at that site. Thus, the straw man ignores LD when
imputing missing data, and the size of the improvement of each method over the straw-
man indicates the effectiveness with which that method exploits patterns of LD in its
imputation algorithm.

SeattleSNPs

To assess the accuracy of the algorithms for imputing missing alleles, I used genotype
data of 24 African Americans (AA) and 23 individuals of European descent (ED) from
the Foundation Jean Dausset-Centre d'Etude du Polymorphisme Humain (CEPH) at
50 genes that were randomly selected from the SeattleSNPs web site (September
2003). These genes have been completely resequenced and contain 15 to 230 seg-
regating sites, with an average of 85 per gene. For convenience, I ignored the few
trilalllic SNPs and multisite insertion-deletion polymorphisms in the data.

To assess how well the various haplotyping methods impute missing genotype
data, I introduced into the data 5% artificial missingness (in addition to the 4.6%
native missingness). This missing data pattern is unlikely to capture all aspects of
patterns of missingness in real data, in which variations in DNA quality or molecular
effects can cause some individuals and some sites to have more than their fair share
of missing data. For this reason, absolute accuracy of methods of this test may tend
to be better than for real data. However, the relative performances should be similar.

For each method I analyzed the data in two ways: 1) analyzing the AA and ED
samples separately and 2) analyzing the combined sample of 47 individuals together.
For the best-performing methods (fastPHASE, PHASE, GERBIL, and HaploBlock)
I computed both the overall ("total") error rates, and error rates stratified by sub-
population (Table 3.2).

The top four methods performed comparably, with fastPHASE achieving the low-
est error rate (although differences among fastPHASE, PHASE and HaploBlock are
not statistically significant). Analyzing data from AA and ED combined seems to give
a small decrease in error rate compared with analyzing them separately, illustrating
the relative robustness of the methods to deviations from HWE.

I also assessed robustness of the results from fastPHASE to the number of clusters, $K$ (Table 3.2). For these data, results were relatively robust across the range of $K$ considered here, at least provided $K$ is sufficiently large (say at least 6). Error rates generally tended to decrease as $K$ increased from 4 to 12, although for sufficiently large $K$ I would expect the error rates to increase, and the rather small differences between $K = 8, 10$ and 12 suggest that larger values of $K$ would not produce a substantial improvement in performance.

The results for fastPHASE in Table 3.2 were all obtained by averaging over parameter estimates from $T = 20$ starts of the EM algorithm. I found that estimates based on only the parameter value (among these 20 estimates) that maximized the likelihood were consistently less accurate. For example, in the “combined” analysis with $K = 6$, this latter (“maximization”) strategy gave an error rate of 0.051, compared with 0.045 for the averaging strategy.

**CEPH HapMap Data**

I obtained HapMap data for chromosomes 7 (41,018 SNPs over 159 Mb) and 22 (15,532 SNPs over 35 Mb) from parents of thirty CEPH trios (60 unrelated individuals) from a Phase I data freeze (March, 2005, http://hapmap.org). I produced datasets with missing genotypes by masking 10% and 25% of genotypes at random and computed error rates. Using fastPHASE I was able to analyze the complete data for each chromosome. However, other methods struggled computationally, and so to allow comparisons with GERBIL and PHASE, I also split the datasets into non-overlapping segments, each containing 150 consecutive SNPs, and analyzed each segment separately. Due to the amount of computation required, I applied PHASE to the chromosome 22 datasets only. (HaploBlock was unable to complete its runs after weeks of computing time.)

Results are given in Table 3.3. For these data, fastPHASE again produced a lower
Table 3.2: Error rates for estimating missing genotypes: SeattleSNPs. Error rates are based on estimating 9479 missing genotypes. AA = Error rate on African American sample, ED = Error rate on European Descent sample, Total = Error rate on combined sample. The best-performing method in each column is highlighted in bold. Differences among fastPHASE, PHASE, and HaploBlock are not statistically significant; differences between these and the other methods are significant (p-value < .007 for contrasts with GERBIL) based on bootstrap resampling of the 50 genes. For fastPHASE, K was selected independently for each analysis (AA only, ED only, and combined).

<table>
<thead>
<tr>
<th>Method</th>
<th>Analyze Separately</th>
<th>Analyze Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AA</td>
<td>ED</td>
</tr>
<tr>
<td>fastPHASE</td>
<td>.053</td>
<td>.024</td>
</tr>
<tr>
<td>PHASE v2</td>
<td>.058</td>
<td>.030</td>
</tr>
<tr>
<td>GERBIL</td>
<td>.067</td>
<td>.030</td>
</tr>
<tr>
<td>HaploBlock</td>
<td>.053</td>
<td>.026</td>
</tr>
<tr>
<td>Bayes-Dirichlet</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>PL-EM</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>HAP</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>hap2</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>snphap</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Straw man</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>fastPHASE : fixed K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K = 4</td>
<td>.066</td>
<td>.029</td>
</tr>
<tr>
<td>K = 6</td>
<td>.059</td>
<td>.024</td>
</tr>
<tr>
<td>K = 8</td>
<td>.058</td>
<td>.026</td>
</tr>
<tr>
<td>K = 10</td>
<td>.054</td>
<td>.027</td>
</tr>
<tr>
<td>K = 12</td>
<td>.053</td>
<td>.026</td>
</tr>
</tbody>
</table>
error rate than the other methods, with the accuracy being slightly better when all
data are analyzed simultaneously rather than in 150-SNP segments. PHASE performs
about as well as fastPHASE, and the improvement of these methods over GERBIL
is more substantial than in the SeattleSNPs data. The main differences between the
datasets are i) the HapMap SNPs tend to have a higher minor-allele frequency, due
to the way in which these SNPs are ascertained, and ii) the HapMap SNPs are more
widely spaced, and so would be expected to exhibit less linkage disequilibrium. Both
these factors presumably contribute to the slightly worse performance of all methods
on the HapMap data, compared with the ED sample of the SeattleSNPs data.

The high accuracy with which genotypes could be estimated with even 25% missing
prompted a more detailed examination of the relationship between accuracy and rates
of missingness (Table 3.4). Although this missing-at-random pattern is not a realistic
assumption for missing data observed in real studies, the fact that accuracy remains
high (> 93%) even with 50% of the genotypes deleted illustrates both the effectiveness
of the methodology and the strong correlations that exist among SNPs at this density.

3.4.2 Haplotype Inference

X-chromosome Data

I assessed accuracy of haplotype estimates using the X-chromosome data from Lin
et al. (2002), also analyzed by Stephens & Donnelly (2003). The data consist of X-
chromosome haplotypes derived from 40 unrelated males. The haplotypes comprise
eight regions, which range in length from 87-327 kilobases and contain 45-165 SNPs.
Each gene consists of 20 pseudo-individuals, created by randomly pairing the 40
chromosomes.

As in previous comparisons of this type, haplotype estimates obtained using different
methods were scored using the following two error rates: the individual error (the
proportion of ambiguous individuals whose haplotypes are not inferred completely
Table 3.3: Error rates for estimating missing genotypes: CEPH HapMap data. For each chromosome, 10% and 25% of the data were masked, resulting in 242,481 and 606,985 missing genotypes for chromosome 7, and 93,476 and 232,731 missing genotypes for chromosome 22. Results for fastPHASE were obtained from 20 random starts of the EM algorithm, except for the “1 start” case, where results were obtained from a single random start.

<table>
<thead>
<tr>
<th>Method</th>
<th>Chr7 (10%)</th>
<th>Chr7 (25%)</th>
<th>Chr22 (10%)</th>
<th>Chr22 (25%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastPHASE</td>
<td>.034</td>
<td>.041</td>
<td>.033</td>
<td>.039</td>
</tr>
<tr>
<td>fastPHASE (1 start)</td>
<td>.046</td>
<td>.057</td>
<td>.045</td>
<td>.056</td>
</tr>
</tbody>
</table>

(whole chromosome)

<table>
<thead>
<tr>
<th>Method</th>
<th>Chr7 (10%)</th>
<th>Chr7 (25%)</th>
<th>Chr22 (10%)</th>
<th>Chr22 (25%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastPHASE</td>
<td>.036</td>
<td>.044</td>
<td>.035</td>
<td>.042</td>
</tr>
<tr>
<td>PHASE v2</td>
<td>-</td>
<td>-</td>
<td>.038</td>
<td>.049</td>
</tr>
<tr>
<td>GERBIL v1.1</td>
<td>.056</td>
<td>.077</td>
<td>.054</td>
<td>.073</td>
</tr>
</tbody>
</table>

(150-SNP datasets separately)

Table 3.4: Error rates for estimating missing genotypes with fastPHASE: CEPH HapMap data, chromosome 22. I masked 10-80% of the data at random, resulting in 93,476 to 837,853 missing genotypes, and applied fastPHASE to the entire chromosome.

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastPHASE</td>
<td>.033</td>
<td>.037</td>
<td>.042</td>
<td>.051</td>
<td>.064</td>
<td>.089</td>
<td>.137</td>
<td>.227</td>
</tr>
<tr>
<td>Straw Man</td>
<td>.381</td>
<td>.384</td>
<td>.382</td>
<td>.386</td>
<td>.387</td>
<td>.391</td>
<td>.395</td>
<td>.403</td>
</tr>
</tbody>
</table>
correctly) and the *switch* error (proportion of heterozygote genotypes that are not correctly phased relative to the previous heterozygote genotype).

Results are given in Table 3.5. For data sets like these, containing at least a moderate number of SNPs, estimating any individual’s haplotypes completely correctly is difficult. As a result, individual error rates of all methods are high and it could be argued that the switch error is a more meaningful measure of performance. However, the qualitative conclusions are the same based on either error rate. Consistent with the results in Kimmel & Shamir (2005), PHASE slightly outperforms GERBIL, which slightly outperforms HaploBlock; fastPHASE produces an individual error rate between that of PHASE and GERBIL, and the lowest switch error (although the difference from PHASE is small and not statistically significant).

As one might hope, the point estimate \( \hat{d}^{SW} \) from fastPHASE, which aims to minimize the switch error, produces a lower switch error than \( \hat{d}^{IN} \), which aims to minimize the individual error. Conversely, \( \hat{d}^{IN} \) produces a lower individual error rate. Results from fastPHASE are again robust to a range of values of \( K \).

For these data, averaging results over multiple parameter estimates obtained from multiple starts of the EM algorithm turns out to be particularly important to obtain good performance. For example, for \( K = 4 \), the point estimate \( \hat{d}^{SW} \) obtained from using only the single set of parameter estimates that give the largest likelihood gives error rates that are worse than any of the other methods considered here (individual error = .716; switch error = .134).

It is also notable that, for these data, even the simple cluster model (which can be obtained as a special case of my model by setting \( r_m = 0 \) for all \( m \)) performs similarly to GERBIL, with individual error of .648 and a switch error of .119. This is perhaps due to lower levels of historical recombination for these X-chromosome genes, and so these data may not provide the best guide to performance of haplotype inference methods on autosomal data.
Table 3.5: Individual and switch error rates for haplotype estimates produced by different methods for the X-chromosome data. The best-performing method in each column is highlighted in bold. The results for Haplotyper are taken from Lin et al. (2002). PL-EM and snphap failed to complete runs for the majority of simulations. For fastPHASE, PHASE (default), GERBIL and HaploBlock, the proportions (error rates) are computed by summing the numerator and denominator over all 8 × 100 datasets; whereas for all the others, proportions are an average of the error rates for each of the 8 genes. For fastPHASE, K was chosen using a subset of the data and then fixed for all 800 data sets. *Computed with only partial data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Individual error</th>
<th>Switch error</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastPHASE (d̂NW)</td>
<td>.649</td>
<td>.110</td>
</tr>
<tr>
<td>fastPHASE (d̂N)</td>
<td>.643</td>
<td>.118</td>
</tr>
<tr>
<td>PHASE v2.1.1</td>
<td>.624</td>
<td>.113</td>
</tr>
<tr>
<td>PHASE v2 (-MQ)</td>
<td>.62</td>
<td>.12</td>
</tr>
<tr>
<td>PHASE V2 (-MS)</td>
<td>.63</td>
<td>.13</td>
</tr>
<tr>
<td>GERBIL v1.0</td>
<td>.660</td>
<td>.118</td>
</tr>
<tr>
<td>HaploBlock</td>
<td>.702</td>
<td>.122</td>
</tr>
<tr>
<td>Bayes–Dirichlet</td>
<td>.68</td>
<td>.15</td>
</tr>
<tr>
<td>Arlequin (ELB)</td>
<td>.66</td>
<td>.18</td>
</tr>
<tr>
<td>hap2</td>
<td>.74</td>
<td>.15</td>
</tr>
<tr>
<td>PL-EM</td>
<td>.73*</td>
<td>.15*</td>
</tr>
<tr>
<td>snphap</td>
<td>.75*</td>
<td>.21*</td>
</tr>
<tr>
<td>Haplotyper</td>
<td>.75</td>
<td>.20</td>
</tr>
<tr>
<td>HAP</td>
<td>.90*</td>
<td>.29*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fastPHASE (Fixed K; d̂NW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K = 4</td>
</tr>
<tr>
<td>K = 6</td>
</tr>
<tr>
<td>K = 8</td>
</tr>
<tr>
<td>K = 10</td>
</tr>
<tr>
<td>K = 12</td>
</tr>
</tbody>
</table>
CEPH HapMap Data

Marchini et al. (2006) compared several methods for haplotype inference on samples of both unrelated and related individuals (trios of parents and child). I consider here the data they used on unrelated individuals, which consist of 3 simulated datasets and 1 real dataset. The simulated data consist of three “Trials” (Trials 1, 2, and 4 from Marchini et al., 2006), which were simulated using coalescent methods with the following conditions, respectively: constant-sized population and constant recombination; constant sized population with variable recombination (a hotspot model); demography approximating that of a European population with variable recombination, with 2% of genotypes masked. The real data consist of unrelated CEPH samples (60 parents from 30 trios) from the HapMap project, where the real haplotypes were determined using the trio data, assuming no recombination from parents to offspring. (Under this assumption the trio data determine phase at a large proportion of sites; the remaining ambiguous sites were ignored when scoring methods.)

I applied fastPHASE to the unphased genotypes from these data and sent the estimated haplotypes ($\hat{\theta}_w$) to J. Marchini, who independently scored the results. Table 3.6 compares the results from fastPHASE with those of other methods in the original comparison. The results from fastPHASE are consistently worse than those of PHASE (and of wphase for the simulated data) and consistently better than those from the other methods, HAP and hap2. Encouragingly for fastPHASE, its performance gap with PHASE is the smallest for the real data, with switch errors of 0.055 vs 0.051.

3.5 Calibration of Predictive Probabilities

All of the comparisons above are based on assessing accuracy of point estimates of estimated genotypes or inferred haplotypes. However, using the models of Chapter 2 it is also quick and easy to compute probabilities for each missing genotype and
Table 3.6: Accuracy of haplotype inference methods for simulated data (Trials 1, 2 and 4) and real data from the HapMap project. Results for fastPHASE were obtained with the ($\hat{e}^{PH}$) estimator. Results for wphase (N Patterson, personal communication), HAP and hap2 were obtained by Marchini et al. (2006). Ind = individual error rate; Switch = switch error rate; Imp = genotype imputation error rate.

<table>
<thead>
<tr>
<th>Method</th>
<th>Trial 1</th>
<th>Trial 2</th>
<th>Trial 4</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ind</td>
<td>Switch</td>
<td>Ind</td>
<td>Switch</td>
</tr>
<tr>
<td>fastPHASE</td>
<td>.653</td>
<td>.045</td>
<td>.887</td>
<td>.069</td>
</tr>
<tr>
<td>PHASE v2</td>
<td>.355</td>
<td>.024</td>
<td>.404</td>
<td>.022</td>
</tr>
<tr>
<td>wphase</td>
<td>.480</td>
<td>.037</td>
<td>.521</td>
<td>.037</td>
</tr>
<tr>
<td>HAP</td>
<td>.886</td>
<td>.065</td>
<td>.971</td>
<td>.098</td>
</tr>
<tr>
<td>hap2</td>
<td>.735</td>
<td>.069</td>
<td>.990</td>
<td>.151</td>
</tr>
</tbody>
</table>

to produce samples from the conditional distribution of haplotypes given the unphased genotypes. These can be used to take account of uncertainty in estimated genotypes/haplotypes in downstream analysis, for example by performing the subsequent analysis on multiple sampled haplotype reconstructions to check for robustness of conclusions, or more formally by using Bayesian statistical methods. However, to justify this strategy, the model should, ideally, produce approximately *calibrated* predictions. For example, of genotypes assessed a probability of 0.9 of being correct, approximately 90% should actually be correct.

I therefore assessed the calibration of predictions from my model, for both genotype imputation and haplotype inference. For genotype imputation, for each imputed genotype I computed the probability, $p_i$ under my model that the imputed genotype was correct (as in section 3.2). I then grouped imputed genotypes into bins, according to their value for $p$, and for each bin I compared the average value of $p$
Figure 3.1: *Calibration of the model for predicting uncertainty in inferred genotypes and haplotypes.* Points (△) represent probabilities obtained by averaging over the 20 runs of the EM algorithm as described in the text.

with the proportion of genotypes that were actually correct. For haplotype reconstruction I examined the calibration of a sample of diplotypc configurations, ≅, by looking at potential switch errors, i.e. by examining, in each individual, the phase of each heterozygous site relative to the previous heterozygous site. For each such pair of heterozygous sites I computed the proportion, q, of the configurations in ≅ that contained the more common of the two possible phasings. I then grouped these site-pairs into bins, according to their value for q, and for each bin I compared the average value of q with the proportion of phasings that were actually correct.

The results (Figure 3.1) show that my model is reasonably well calibrated, but slightly conservative, for both tasks. For example, of genotypes assessed a 90% chance of being correct, roughly 96% were actually correct in both the SeattleSNPs and HapMap data. One curious feature of the results is the slight drop in accuracy for the highest-confidence bin (corresponding to 98% predicted probability of being correct, this limit being due to the limits of 0.99 and 0.01 I imposed on elements of θ) in the SeattleSNPs data. Closer inspection reveals that this is due to errors
in imputing genotypes of masked heterozygotes at singleton SNPs (i.e., SNPs where only one copy of the minor allele is present). When such genotypes are masked, fastPHASE very confidently, but wrongly, assesses them to be homozygous for the only observed allele. This could be viewed as an artifact, due to the fact that the data contain only polymorphic SNPs, and this model does not condition on the markers being polymorphic.

3.6 Computational Requirements

The relative computation times of the different methods I consider here vary across sets of data and depend on the way the methods are applied (e.g., how many iterations were used). As I applied the methods, fastPHASE and GERBIL require similar amounts of computational resources, and both are considerably faster than PHASE and HaploBlock. Computation times are summarized for all the results in Table 3.7. The speed-up for fastPHASE, compared with PHASE, would be greater for samples with a larger number of individuals.

3.7 Remarks

Missing genotype imputation and haplotype inference applications are considered again in Chapter 5, applying a model which allows for structure to many of the same data sets presented above. These methods lay the groundwork for using this model for applications which require completely-observed data or observed haplotypes. For such applications, samples of these desired quantities, given the observed unphased genotypes, can be drawn repeatedly, and methods may then be applied to these sampled quantities of interest, appropriately combining results from each sample.

This multiple imputation is in fact the framework for an application to whole-genome association studies, which is outlined in Chapter 7 (see “Future Work”). This whole-genome association scan motivates additional considerations of the way parameters of this model are estimated, and these issues are addressed in the following
Table 3.7: Comparison of computation times (in hours) for different methods. Each method was applied on a 3 GHz Xeon processor with 1 GB of memory. The SeattleSNPs column is for analysis of all 47 individuals together. For estimating missing genotypes, fastPHASE was applied to the data without inferring haplotypes. Times for HaploBlock on the SeattleSNPs data and PHASE v2 on HapMap chromosome 22 are based on extrapolation from a subset of the data sets. Approximate amounts of missing data are in parentheses. The fastest method for each data set is highlighted in bold. Hap. inference = haplotype inference.

<table>
<thead>
<tr>
<th>Missing genotype estimation</th>
<th>SeattleSNPs (10%)</th>
<th>HapMap Chr. 7 (10%)</th>
<th>HapMap Chr. 22 (25%)</th>
<th>HapMap Chr. 22 (10%)</th>
<th>HapMap Chr. 22 (25%)</th>
<th>X-chromosome (7%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastPHASE</td>
<td>1.3</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>PHASE v2</td>
<td>29.3</td>
<td>n/a</td>
<td>n/a</td>
<td>323</td>
<td>720</td>
<td>151</td>
</tr>
<tr>
<td>GERBIL</td>
<td>0.6</td>
<td>29</td>
<td>32</td>
<td>10</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>HaploBlock</td>
<td>160.1</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>265</td>
</tr>
</tbody>
</table>
chapter.
Chapter 4

IMPOSING PARAMETRIC STRUCTURE

In this chapter I consider reparameterizations of the models presented in Chapter 2. I explore several model variations which impose constraints on parameters of the Markov chain for haplotype-cluster membership and investigate their effect on the missing data and haplotype applications. I also consider the problem of estimating cluster-specific allele frequencies at untyped SNPs or from at most a small number of genotypes at some marker loci.

4.1 Parametric Constraints in the Markov Chain for Cluster Membership

4.1.1 Homogeneous Poisson Process and Constant Cluster Frequency Distribution

In Sections 2.2 and 2.3, a Markov chain, given by (2.9) and (2.10), was introduced to model the clusters-of-origin $z_i$ for each individual’s multilocus genotype data. The estimators for $\alpha$ and $r$ in Appendix C allow these parameters to be estimated independently for each SNP. However, since allowing a separate value of $\alpha$ and $r$ for each SNP results in a large number of parameters, one might wonder whether predictions would be improved by imposing restrictions on some of these parameters. Here, I investigate these possibilities.

A jump in the Markov chain occurs before SNP $m$ ($m = 2, \ldots, M$) if at least one event occurs from a Poisson process of rate $r_m$. The constant $r$ model restricts $r_m = r_{m'}$ ($2 \leq m \leq m' \leq M$). Conditional on a jump “before” marker $m$, a new cluster is chosen at random from $\alpha_m = (\alpha_{1m}, \ldots, \alpha_{Km})$. The constant $\alpha$ model restricts $\alpha_m = \alpha_{m'}$ ($1 \leq m \leq m' \leq M$). The constant $\alpha, r$ imposes both of these
Table 4.1: Error rates for missing data imputation and haplotype estimation using constant parameters of the Markov chain for cluster membership. The “unconstrained” model is the general model for which results are presented elsewhere. All results were obtained from averaging over 20 random starts of the EM. For each data set in the SeattleSNPs (50 data sets) and X-chromosome (800 data sets) data, K was determined separately from among 4, 6, 8, 10 and 12. For each HapMap data set, K was selected among 8, 13 and 18.

<table>
<thead>
<tr>
<th>Missing Data Imputation</th>
<th>Haplotype Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SeattleSNPs Combined</td>
<td>HapMap Chr 21</td>
</tr>
<tr>
<td></td>
<td>CEU</td>
</tr>
<tr>
<td>unconstrained</td>
<td>.037</td>
</tr>
<tr>
<td>constant r</td>
<td>.037</td>
</tr>
<tr>
<td>constant α</td>
<td>.036</td>
</tr>
<tr>
<td>constant α, r</td>
<td>.037</td>
</tr>
</tbody>
</table>

restrictions. Parameter estimation under these constraints is discussed below, at the end of section 4.1.2.

I applied these models to several of the data sets considered previously. Results are given in Table 4.1. From these results, it appears that constraining α to be constant offers a slight improvement in performance over other parameterizations. The difference is the greatest for haplotype estimation of pseudo-individuals composed of X chromosomes, particularly in terms of individual error. Other differences were small.

The SeattleSNPs and X chromosome data are dense and cover short genomic regions, with a typical data set consisting of 100 SNPs over 20 kb of DNA sequence. The density of the HapMap chromosome is approximately 1 SNP per kb and, for chromosome 2, covers 240,000 kb. An assumption of constant α seems more reasonable for
data from smaller genomic regions and may be beneficial in regions of high LD where there are fewer "jumps" in the Markov chain, as these jumps provide information for the estimation of $\alpha$. Due to the large number of parameters in this model, especially with models in Chapter 6, an assumption of constant $\alpha$ may provide a convenient and efficient way to reduce the number of parameters which need to be estimated. However, it is slightly surprising that the unconstrained model does not outperform the "constant $\alpha$" model in the application to chromosome 2 (nor does the "constant $r$" model outperform the "constant $\alpha, r$" model).

It is not surprising that the "constant $r$" models do slightly worse than allowing $r$ to vary across the chromosome. Physical distance may have little relationship with the amount of LD in many regions, and these constant-$r$ models assume a consistent relationship. These considerations motivated the exploration of the parameterizations below.

4.1.2 Ad hoc Smoothing

Additional parameterizations of the Markov chain are also possible. A compromise between the unconstrained and constant-parameter models is one with "locally-smooth" parameters; that is, $\alpha$ and $r$ are allowed to change gradually along the chromosome.

I considered two ad hoc approaches to obtaining smoothed parameter estimates of $\alpha$ and one for $r$. The first for $\alpha$ is to use pseudocounts. The ML estimator for $\alpha$ is given by (C.1):

$$\hat{\alpha}_{km} \propto \sum_{i=1}^{n} E_{i,r}[j_{imk}|g],$$

with $\sum_{k=1}^{K} \alpha_{km} = 1$. An alternate estimator based on pseudocounts is given by the following:

$$\bar{\alpha}_{km} \propto \sum_{i=1}^{n} E_{i,r}[j_{imk}|g] + \delta,$$

with $\sum_{k=1}^{K} \alpha_{km} = 1$. I considered $\delta = 1$. The pseudocounts will make the distribution of $\alpha_{km}$ more uniform, with this effect most pronounced at sites where $\sum_{i} \sum_{k} j_{imk}$ is
small.

A second strategy for parameter-smoothing is a sliding-window approach, which combines the information from \( W \) SNPs on either side of \( m \) for estimation of \( \alpha_{km} \). A parameter \( \omega \) controls the relative contribution of information from SNPs based on their proximity to SNP \( m \). This is given by:

\[
\bar{\alpha}_{km} = \sum_{s=-W}^{m+W} \sum_{i=1}^{n} \left( E_{\theta^*} \left[ j_{isk} | g \right] \right) \omega^{m-s},
\]

for \( W < m \leq M - W \), with \( \sum_{k=1}^{K} \alpha_{km} = 1 \). From (C.2), I have

\[
\hat{r}_m = -\log \left( \frac{1 - \sum_{k=1}^{K} E_{\theta^*} \left[ j_{isk} | g \right]}{d_m} \right).
\]

A sliding-window estimator for \( r \) is:

\[
\bar{r}_m = -\log \left( \frac{1 - \sum_{s=m-W}^{m+W} \sum_{k=1}^{K} E_{\theta^*} \left[ j_{isk} | g \right]}{d_{m+W} - d_{m-W-1}} \right)
\]

for \( W + 1 < m \leq M - W \).

Parameter estimators for the constant-\( \alpha \) and constant-\( r \) models given above in (4.1.1) are essentially special cases of those for \( \bar{\alpha} \) and \( \bar{r} \) above, with \( \omega = 1 \) for \( \bar{\alpha} \). The sums indexed by \( s \) are over all sites for \( \bar{\alpha} \) and from 2 to \( M \) for \( \bar{r} \).

To test these estimation approaches, I applied the model to the SeattleSNPs data for the purposes of imputing missing genotypes, considering various combinations of the estimators \( \bar{\alpha} \), \( \bar{\theta} \), and \( \bar{r} \), with \( W = 1 \), 3 and 5, and \( \omega = 1 \), .8 and .5. I used these estimators in place of the MLEs in the EM algorithm of Appendix C. There is no formal justification for this procedure. The likelihood is not guaranteed to be non-decreasing between successive iterations. The results in terms of missing genotype imputation for the SeattleSNPs data were typically within the range of values presented in Table 4.1 (results not shown).
4.2 Estimation of Cluster-Specific Allele Frequencies at Untyped SNPs

Now consider the situation where at most a few genotypes are observed at some SNP markers. At such a SNP, estimates of cluster-specific allele frequencies $\theta$ may be poor. In such cases, one might hope to improve the estimation of $\theta$ using information from "typical" values of $\theta$ observed at other SNPs, both genome-wide and locally. Here I investigate this possibility.

4.2.1 Motivation and Strategy

Let $x_m$ denote the true genotypes that were unobserved at $m$, and let $g_m$ denote any observed genotype data at site $m$, so that $g = (g_m, g_{\neg m})$. The goal is to impute $x_m$, or learn about its distribution, using all observed data $g$.

Consider first the situation where no genotypes are observed at SNP marker $m$. If all the parameters of the model were known, the distribution of genotypes at $m$ would also be known, given $g$, via equation (3.1). The problem is that if no genotypes are observed at $m$, then estimating parameters ($\alpha_m$, $\theta_m$, $r_m$) associated with that SNP is not straightforward, particularly for $\theta_m$. Here, I propose a solution to this problem.

First, assume that parameters at other SNPs, ($\alpha_{\neg m}$, $\theta_{\neg m}$, $r_{\neg m}$), have been obtained via an EM algorithm (Appendix C). As shown in section 4.1, estimates for $\alpha$ and $r$ may be smoothed across large regions with little or no decrease in performance (for imputing missing genotypes). This motivates the following estimators for $r_m$ and $\alpha_m$:

\[
\hat{r}_m = \frac{\hat{r}_{m-1} (d_m - d_{m-1}) + \hat{r}_{m+1} (d_{m+1} - d_m)}{d_{m+1} - d_{m-1}},
\]

\[
\hat{\alpha}_{km} = \frac{\hat{\alpha}_{k(m-1)} (d_m - d_{m-1}) + \hat{\alpha}_{k(m+1)} (d_{m+1} - d_m)}{d_{m+1} - d_{m-1}},
\]

for $k = 1, \ldots, K$ and $m = 2, \ldots, M - 1$ for $\alpha$ and $m = 3, \ldots, M - 1$ for $r$.

It is less straightforward to estimate cluster-specific allele frequencies $\theta$ at sites without observed data. If $\theta$ had a known distribution, one could obtain the probability
of genotypes at such sites by integrating out $\theta$. The probability distribution for the unobserved data $x_m$ could then be obtained by:

$$
p(x_m|g, \hat{\theta}_{(-m)}, \hat{\alpha}, \hat{r}) = 
\int p(x_m, \theta_m|g, \hat{\theta}_{(-m)}, \hat{\alpha}, \hat{r}) d\theta_m = 
\int p(x_m|g, \theta_m, \hat{\theta}_{(-m)}, \hat{\alpha}, \hat{r}) \pi(\theta_m) d\theta_m.
$$

(4.1)

Of course, $\pi$ is unknown. However, if I could find a suitable approximation $\hat{\pi}$, from which I could sample values for $\theta_m$, this would suggest the following approximation to (4.1):

$$
p(x_m|g, \hat{\theta}_{(-m)}, \hat{\alpha}, \hat{r}) \approx \frac{\sum_{i=1}^{B} \prod_i p(x_{im}|g_i, \theta_{m}^{(b)}, \hat{\theta}_{(-m)}, \hat{\alpha}, \hat{r})}{B},
$$

(4.2)

where $\theta_{m}^{(1)}, \ldots, \theta_{m}^{(B)}$ is a random sample from $\hat{\pi}$ and $B$ is some large number. To calculate $p(x_{im}|g_i, \theta_{m}^{(b)}, \theta_{(-m)}, \alpha, r)$, I sum over possible values of $z_{im}^*$:

$$
\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(x_{im}|z_{im}^* = \{k_1, k_2\}, \theta_m)p(z_{im}^* = \{k_1, k_2\}|g_i, \theta_{(-m)}, \alpha, r).
$$

4.2.2 Potential Approximations for $\pi$

My strategy for approximating $\pi(\theta_m)$ is to use the estimated values of $\theta$ at all other SNPs, i.e. the columns of $\hat{\theta}_{(-m)}$, as the support of $\hat{\pi}$, and to weight these values in different ways. Specifically, I consider the following approximations:

a. Uniform on all columns of $\hat{\theta}_{(-m)}$, $\hat{F}_\theta$

b. Uniform distribution on $\{\hat{\theta}_s : m - W \leq s \leq m + W\}$, $\hat{F}_\theta^W$

c. A weighted average of these: $u \hat{F}_\theta^W + (1 - u) \hat{F}_\theta$, $0 < u < 1$. 
I explored values of 5, 10, 20 and 40 for \( W \), and \(.2, .4, .6 \) and \(.8 \) for \( u \).

In addition to the distributions above, I considered a variation on these, constructed by the following brief algorithm:

1. Sample a column \( \theta_m \) from \( \hat{\theta}_{(-m)} \) according to one of the above distributions.

2. Replace elements of \( \theta_m = (\theta_{m_1}, \ldots, \theta_{m_K}) \) that are in \((.02,.98)\) with uniform draws from the elements of \( \hat{\theta}_{(-m)} \) which were between \(.02\) and \(.98\).

The motivation for this additional “nonboundary” strategy comes from the observation that most elements of \( \hat{\theta} \) are near 0 or 1 when the genetic variation in a sample is being captured well by the model. Estimated elements that are not near 0 or 1, and thus not capturing LD as well, may have less relevance at other sites. I wanted to consider a distribution which represented the uncertainty in these estimates.

The proposed approximations for \( \pi(\theta_m) \), \( b \) and \( c \), give more weight to columns of \( \hat{\theta}_{(-m)} \) in closer proximity to the (untyped) SNP at \( m \). This strategy was motivated by an empirical observation that there is local structure in estimated values of \( \theta \) from the model. This is evidenced in Figure 4.1.

The top half of Figure 4.1 is a visual representation of the vectors of \( \theta \) estimated at 81 adjacent SNPs from chromosome 22 of the HapMap (60 unrelated individuals from CEU panel). There is considerable similarity of vertical columns among nearby SNPs. This is perhaps most apparent in the region from 10,045 to 10,065. The vectors of estimates are identical at almost every cluster from approximately 10,003 to 10,009, as well. Additionally, at many pairs of adjacent sites, the columns are mirror images of each other. That is, every white square at a cluster in one column is black in the same cluster at the other column (and \textit{vice versa}). At each SNP, the two allelic types are arbitrarily ordered, and so these mirror-images are the result of equivalent distributions for values of \( \theta \) among the clusters. That is, the mirror images represent elements of the same \textit{equivalence class}. Because of this, \( \log q(G) \) is calculated as a sum over the two members of an equivalence class.
Figure 4.1: Representations of elements of $\theta$. Gray-scale representation of the values of $\theta$ for a 80-SNP region of CEU chromosome 22 of the HapMap. Values near one are black; values near zero are white. Cluster-specific allele frequencies are in columns; rows correspond to allele frequencies of clusters. TOP: $\theta$ values for SNP 10,000 to 10,080, order consistent with the data. BOTTOM: Columns in top plot are randomly permuted.
For comparison, the bottom plot is the representation resulting from a random permutation of the columns in the top. This difference would be more extreme over a larger region, as the local nature of this structure can cover up to 40 SNPs, on the order of that depicted here.

4.2.3 Evaluating Approximations for \( \pi \)

I now turn to the question of which of the above choices for \( \hat{\pi} \) (\( a \), \( b \), or \( c \)), when used in (4.2), provides the “best” approximation to the “true” distribution of genotypes. One way to measure the quality of an approximate distribution \( q \) for a true distribution \( p \) is with the Kullback–Leibler (K-L) information for discrimination between two densities. Let \( G \) denote untyped genotypes at a SNP. The “K-L distance” is given by:

\[
\int \log \frac{p(G)}{q(G)} p(G) dG,
\]

Among the approximations presented above, I seek a \( q \) that minimizes this distance or, equivalently, maximizes

\[
\int \log [q(G)] p(G) dG. \tag{4.3}
\]

Of course, I do not know the true distribution \( p \) but can take observed genotypes at each SNP as a sample from \( p(G) \) and, using (4.2) for \( q \), approximate (4.3) as

\[
\sum_{d \in D} \log \left[ \frac{\sum_{i=1}^{n} \prod_{k} p(x_{id} | \theta_{d}^{(k)} \bar{x}_{d-1} \bar{y}_{d} \bar{z}_{d})}{|D|} \right],
\]

where \( D \) is a set of indices sampled uniformly from \( \{1, \ldots, M\} \) and for each \( d \) of these \( |D| \) samples, \( \theta_{d}^{(1)}, \ldots, \theta_{d}^{(B)} \) is a random sample from one of the distributions considered above, which may depend on \( d \).

Table 4.2 contains estimated means of \( \log q(G) \) for several of the distributions considered for approximation of \( \pi \). The results are consistently better for strategies that use a combination of local and long-range sampling. Approximately equal
Table 4.2: Sample means of log q(G) for several distributions. log q(G) was calculated at 952 sites using multiple approximations \( \hat{\pi} \). Several of these are given here. The 952 values consisted of results from 6 random starts of the EM algorithm, with \( K = 12 \).

<table>
<thead>
<tr>
<th></th>
<th>Max. distance from SNP (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>( \hat{\pi}_0 )</td>
<td></td>
</tr>
<tr>
<td>( \hat{\pi}_0^{\hat{\pi}} )</td>
<td>-16.5</td>
</tr>
<tr>
<td>( 2\hat{\pi}_0^{\hat{\pi}} + 0.8\hat{\pi}_0 )</td>
<td>-14.4</td>
</tr>
<tr>
<td>( 0.6\hat{\pi}_0^{\hat{\pi}} + 0.4\hat{\pi}_0 )</td>
<td><strong>-14.2</strong></td>
</tr>
<tr>
<td>( 0.6\hat{\pi}_0^{\hat{\pi}} + 0.4\hat{\pi}_0 ) (nonboundary)</td>
<td>-14.5</td>
</tr>
</tbody>
</table>

weightings of these strategies did relatively well (results were best for the \( 0.6\hat{\pi}_0^{\hat{\pi}} + 0.4\hat{\pi}_0 \) distribution).

4.2.4 Application to Imputing Missing Genotypes

To test the methodology developed in Section 4.2.1, I attempt to impute missing genotypes, comparing this method (“\( \hat{\theta} \)-integration”) to the “maximum likelihood” (ML) approach used in Chapter 3. Since the ML approach requires some data at a SNP with which to estimate the parameters, I consider the situation where only a few genotypes are observed at some SNPs.

In obtaining the last line of (4.1), I used the assumption that there were no genotypes observed at the marker of interest \( m \). Thus \( p(\theta_m|g, \hat{\theta}_{(\neg m)}, \hat{\alpha}, \hat{\rho}) \) was rewritten as the marginal distribution for the \( \theta \) values at a site: \( \pi(\theta) \). Here, since there are nonmissing genotypes at \( m \), I have:

\[
p(\theta_m|g, \hat{\theta}_{(\neg m)}, \hat{\alpha}, \hat{\rho}) \propto p(g|\theta_m, \hat{\theta}_{(\neg m)}, \hat{\alpha}, \hat{\rho})\pi(\theta_m).
\]

For this application the strategy for both methods is to impute the most-likely geno-
Table 4.3: *Genotype estimation error rates from HapMap chromosome 22*. Genotypes at 778 sites from chromosome 22 were masked at the rates of .5 (23,070 missing genotypes) and .85 (39,613 missing genotypes), and then estimated with the above methods. Results are from 5 random starts of the EM algorithm, with $K = 5$ and 25 iterations. “$\theta$-integration” averages are from 3,000 samples (plus 3,000 of equivalence class), with $W = 10$ and $u = .5$.

<table>
<thead>
<tr>
<th>% Masked at a SNP</th>
<th>50%</th>
<th>85%</th>
</tr>
</thead>
<tbody>
<tr>
<td>“ML”</td>
<td>.050</td>
<td>.097</td>
</tr>
<tr>
<td>$\theta$-integration</td>
<td>.060</td>
<td>.089</td>
</tr>
</tbody>
</table>

types, and so the constant-of-proportionality may be ignored. For other applications, a probability may be obtained by considering a large number and normalizing or by calculating $p(g|\hat{\theta}_{(-m)}, \hat{\alpha}, \hat{\pi})$ directly.

At 778 sites, I masked 50% and 85% of genotypes on chromosome 22 of 60 unrelated CEU individuals from Phase I of the HapMap. Genotypes were imputed by both methods, using the unmasked data at these sites and all other observed genotypes. Results are in Table 4.3.

The “$\theta$-integration” method of this section outperforms the ML method when enough data (85%) are missing. However, at levels of missingness of 50%, the ML method is superior. It seems that one might be able to outperform the ML method when half the genotypes are missing by borrowing strength from neighboring regions. That the $\theta$-integration method does not perform as well as the ML method in this setting may reflect a suboptimal choice of $\hat{\pi}$ or insufficient samples; however, in Chapter 3 I demonstrated that the ML method performs quite well at estimating missing genotypes even in the presence of large amounts of missing data, and therefore the ML results may be difficult to achieve by other methods. Further examination of these issues, including different approximations for $\pi$, is necessary but has not been
performed at this time.
Chapter 5

POPULATION STRUCTURE

In this chapter I present a model for taking explicit account of population structure when this structure is known. I then demonstrate that this extension can increase the accuracy of the missing data and haplotype estimation applications considered in Chapter 3.

5.1 Incorporation of Population Structure in the Likelihood

The model for genotypes from diploid individuals (2.13) is constructed with the assumption of HWE of the haplotypes from (2.8). Although the assumption of HWE will not hold exactly for real populations, previous studies have consistently suggested that models based on HWE can perform well at haplotype inference and missing data imputation, even when there are clear and substantial deviations from HWE (e.g. Fallin & Schork, 2000, and Chapter 2).

Nevertheless, I examined the potential benefits of modifying the above model to deal with a certain type of deviation from HWE. Specifically I consider the situation where individuals are sampled from \( S \) distinct subpopulations. Let \( s_i \in 1, \ldots, S \) denote the subpopulation of origin for individual \( i \) (assumed known). Due to shared ancestry, different subpopulations may share some of their haplotype structure, although haplotype frequencies and levels of LD would be expected to differ. To capture this, I enforce the \( \theta \) parameters, which characterize the "common haplotypes" in the sample, to be shared across subpopulations but allow the \( \alpha \) and \( \tau \) parameters, which relate to the common haplotype frequencies and rates at which cluster membership changes, to vary among the subpopulations. Thus \( \alpha = (\alpha^{(1)}, \ldots, \alpha^{(S)}) \) and \( \tau = (\tau^{(1)}, \ldots, \tau^{(S)}) \),

\}
where \((\alpha^{(j)}, r^{(j)})\) denote the parameters relating to subpopulation \(j\), and

\[
p(g|\alpha, \theta, r, s) = \prod_{i=1}^{n} p(g_i|\alpha^{(s_i)}, \theta, r^{(s_i)}),
\]

where \(s = (s_1, \ldots, s_n)\) and \(p(g_i|\alpha^{(s_i)}, \theta, r^{(s_i)})\) is as in (2.13).

Parameter estimation is a straightforward application of the procedures given in Appendix C. The parameters of the Markov chain for individuals in subpopulation \(s\) \((\alpha^{(s)}, r^{(s)})\) are estimated using only the data from individuals sampled from subpopulation \(s\) \((s = 1, \ldots, S)\).

### 5.2 Application to Data from Structured Populations

To examine whether accuracy could be improved by taking account of the subpopulations of different samples, I applied the model above to data from SeattleSNPs and The HapMap Project for the purpose of imputing missing genotypes. The HapMap data consist of samples from four “analysis panels”; European/CEPH (CEU), Han Chinese from Beijing (CHB), Japanese from Tokyo (JPT), and Yoruba from Ibadan, Nigeria (YRI). Data from both SeattleSNPs and the HapMap were analyzed in 3 ways: a) explicitly modelling the “subpopulations” as in (5.1); b) analyzing all individuals together, ignoring the sampling design; and c) analyzing each population separately.

The SeattleSNPs data are the same as considered in Section 3.4.1. The HapMap data were obtained from a “Phase II” release (26 September, 2005) of the 290,631 SNPs of chromosome 2. Only the genotypes of unrelated individuals from 4 analysis panels were used for the missing data imputation. Results are in Table 5.1.

Additionally, since genotypes from the HapMap were collected on parent-child trios for the CEU and YRI analysis panels, haplotypes of these individuals were inferred with high accuracy (Marchini et al., 2006). I obtained these haplotypes (inferred from pedigree information) for the unrelated parents of the CEU and YRI panels and created phase-unknown data sets for analysis. As for the missing genotype
Table 5.1: Comparison of approaches to dealing with population structure: missing data estimation and haplotype estimation. Table entries are error rates: proportion of genotypes imputed incorrectly and switch error. Results were obtained from averaging over 20 random starts of the EM algorithm. Choice of $K$ was made over the range of 4-12 for the SeattleSNPs data and 8-18 for HapMap data. There were 9,479 masked genotypes imputed in the SeattleSNPs data and 12,022,516 imputed for HapMap chromosome 2.

<table>
<thead>
<tr>
<th></th>
<th>Genotype Estimation</th>
<th>Haplotype Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SeattleSNPs</td>
<td>HapMap Chr. 2</td>
</tr>
<tr>
<td></td>
<td>(Afr-Am, CEPH)</td>
<td>4 panels</td>
</tr>
<tr>
<td>(% missing)</td>
<td>(10%)</td>
<td>(20%)</td>
</tr>
<tr>
<td>Model structure</td>
<td>.035</td>
<td>.019</td>
</tr>
<tr>
<td>Ignore structure</td>
<td>.037</td>
<td>.020</td>
</tr>
<tr>
<td>Analyze separately</td>
<td>.039</td>
<td>.022</td>
</tr>
</tbody>
</table>

estimation, I analyzed these data 3 ways (together with structure, together ignoring structure, and separately), estimating the haplotypes and comparing the estimates to the pedigree-inferred haplotypes. These comparisons are in the last column of Table 5.1.

For all three data sets, modelling population structure offered improvement over analyzing the groups together but ignoring the structure. For the haplotype inference application, this improvement was consistent across all chromosomes analyzed. The magnitude of the improvement will depend on how genetically distinct the subpopulations are. When genetic differences are pronounced or with sufficiently large sample sizes, there will be little gain in performance from combining all samples together for analysis. However, for other applications or analyses, where all populations share
certain parameters of interest, for example, it may be imperative to analyze all the data at once. In such cases, modelling the population structure may be beneficial.

One example of this situation is when samples are collected at distinct locations for a genetic association study. In this situation it might not be appropriate to consider the entire collection of genotypes as a random sample from a randomly mating population. Such population structure can lead to spurious associations between genotype and phenotype (e.g., Lander & Schork, 1994). A related issue arises in case–control studies of genetic association, even when samples are collected from one geographic location. Analysis of all subjects together may mitigate important genetic differences, whereas analyzing the cases and controls separately may exaggerate differences between the groups. A compromise would be to estimate the common haplotypes ($\theta$) using all the data but to allow separate estimates of these cluster frequencies ($\alpha$) for the two (cases and controls) samples. However, a more detailed investigation of this type of analysis for case–control studies is necessary before the approach can be advocated. Still, the model given in this chapter may provide a tool by which one could account more appropriately for these types of sampling designs.
Chapter 6

GENOTYPING ERROR

In Chapters 2 and 3, I assumed that the genotype data were observed without error but possibly with some genotypes “missing”. Here I extend the model to allow explicitly for genotyping error. I consider several different mechanisms for the occurrence of errors and present algorithms for parameter estimation in this context. I consider the problem of detecting genotype errors in population data based on patterns of LD and present a method for this task. Finally, I perform a scan for errors in data from The International HapMap Project.

6.1 Incorporating Error Parameters into the Likelihood

Again, for simplicity of notation, I assume $K$ is known. Let $g$ continue to denote the observed genotypes, and let $x_{im} \in \{0, 1, 2\}$ denote the true genotype of individual $i$ at SNP marker $m$, with $x$ denoting the matrix of all true genotypes. Then, genotypes $g$ are observed, possibly with error, according to some model $p(g|x, \epsilon)$, where $\epsilon$ represents an error rate (or vector of rates).

As before, $z_i$ denotes the vector of cluster memberships for individual $i$. Conditional on the cluster memberships, the probability of the observed data may be calculated by summing over the possible true genotypes as follows:

$$p(g|z_{im}, \theta, \epsilon) = \prod_{m=1}^{M} \sum_{a=0}^{2} p(g_m|z_{im} = a, \epsilon)p(x_{im} = a|z_{im}, \theta),$$
with

\[
p(x_{im} | z^*_m = \{k_1, k_2\}, \theta) = \begin{cases} 
(1 - \theta_{k_1m})(1 - \theta_{k_2m}), & x_{im} = 0, \\
\theta_{k_1m}(1 - \theta_{k_2m}) + \theta_{k_2m}(1 - \theta_{k_1m}), & x_{im} = 1, \\
\theta_{k_1m}\theta_{k_2m}, & x_{im} = 2. 
\end{cases}
\]

As in Chapter 2, since the cluster memberships are unobserved, I have to sum over all possible \(z_i\):

\[
p(g_i | \alpha, \theta, r, \epsilon) = \sum_{z_i} \prod_{m=1}^{M} \sum_{a=0}^{2} p(g_{im} | x_{im} = a, \epsilon) p(x_{im} = a | z^*_m, \theta) p(z^*_m | \alpha, r). 
\]

As this sum involves a very large number of terms for any realistic data, naive calculation of the likelihood is computationally intractable. However, in Appendix D, I present an efficient algorithm for its calculation based on Baum-Welch algorithms for HMMs. Finally, since conditional on the parameters of the model, genotypes from different individuals are independent, I obtain the likelihood \(L(\alpha, \theta, r, \epsilon; g) = p(g | \alpha, \theta, r, \epsilon)\) with

\[
p(g | \alpha, \theta, r, \epsilon) = \prod_{i=1}^{n} p(g_i | \alpha, \theta, r, \epsilon). 
\]

6.2 Error Models

I consider three error models, described below. In all cases, conditional on the model parameters, errors are assumed to occur independently across sites and across individuals.

**Simple Error Model** In this model, errors are assumed to occur at rate \(\epsilon\), independent of the true genotypes. Conditional on an error occurring, a new genotype is chosen uniformly at random from \(\{0, 1, 2\}\). These assumptions lead to the **simple**
**error model**, given by

\[ p(g_{im} = a | x_{im} = b, \epsilon) = \begin{cases} 
1 - \frac{2}{3} \epsilon, & a = b, \\
\frac{1}{3} \epsilon, & a \neq b, 
\end{cases} \]

\(0 \leq \epsilon \leq 1\), for \(i = 1, \ldots, n\) and \(m = 1, \ldots, M\). The model may also be represented by the “transition probability matrix”

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 - (\frac{2}{3} \epsilon)</td>
<td>(\frac{\epsilon}{3})</td>
<td>(\frac{\epsilon}{3})</td>
</tr>
<tr>
<td>true</td>
<td>(\frac{\epsilon}{3})</td>
<td>1 - (\frac{2}{3} \epsilon)</td>
<td>(\frac{\epsilon}{3})</td>
</tr>
<tr>
<td>2</td>
<td>(\frac{\epsilon}{3})</td>
<td>(\frac{\epsilon}{3})</td>
<td>1 - (\frac{2}{3} \epsilon),</td>
</tr>
</tbody>
</table>

with \(0 \leq \epsilon \leq 1\). Note that under this model the actual errors occur at rate \(\frac{2}{3} \epsilon\). This choice of parameterization is in part due to computational convenience and also for its mathematical property that as \(\epsilon \to 1\), all observed genotypes become equally probable, regardless of the true genotypes.

**Simple 2-Parameter error Model**  The simple model can be generalized to allow for different rates of errors depending on the true genotype, perhaps a more realistic model. Let \(\epsilon = (\epsilon^0, \epsilon^1)\), with \(\epsilon^0\) denoting the probability of an error event if the true genotype is a homozygote and \(\epsilon^1\) denoting the corresponding probability for a heterozygote. This model may be represented by:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 - (\frac{2}{3} \epsilon^0)</td>
<td>(\epsilon^0)</td>
<td>(\epsilon^0)</td>
</tr>
<tr>
<td>true</td>
<td>(\epsilon^1)</td>
<td>1 - (\frac{2}{3} \epsilon^1)</td>
<td>(\epsilon^1)</td>
</tr>
<tr>
<td>2</td>
<td>(\epsilon^0)</td>
<td>(\epsilon^0)</td>
<td>1 - (\frac{2}{3} \epsilon^0),</td>
</tr>
</tbody>
</table>

where \(0 \leq \epsilon^0, \epsilon^1 \leq 1\).
Allelic Error Model  Finally, I describe an error model in terms of alleles, which allows for different rates depending on the true genotype. An allele may “drop out” so that that a true heterozygote is observed as a homozygote. Alternatively, an allele may “drop in” so that a heterozygote is observed when the truth is a homozygote. By assuming at most one allelic error per observed genotype and allowing for different drop-in and drop-out rates, I obtain the following error model, represented as a transition probability matrix:

\[
\begin{array}{c|ccc}
   & 0 & 1 & 2 \\
\hline
0 & 1 - \epsilon^0_2 & \frac{\epsilon^0_1}{2} & 0 \\
true & \frac{\epsilon^1_3}{3} & 1 - \frac{2\epsilon^1_3}{3} & \frac{\epsilon^1_3}{3} \\
2 & 0 & \frac{\epsilon^0_2}{2} & 1 - \frac{\epsilon^0_2}{2}
\end{array}
\]

where \(0 \leq \epsilon^0, \epsilon^1 \leq 1\). By assuming at most one allelic error per observed genotype, this model does not allow for the observation of a homozygote of one allelic type when the truth is a homozygote of the other allele.

More general error models are certainly possible and may be appropriate in some cases. However, here I restrict attention to the above models.

6.2.1 Parameter Estimation

Estimation of parameters of the error models presented above may be accomplished via an EM algorithm, which I present in Appendix D. The error parameters \(\epsilon\) are estimated jointly with \(\alpha, \theta,\) and \(r\). The order of complexity for the algorithm is \(O(K^2 Mn)\).

6.3 Methods for Detection

While considerable attention has been given to the detection and effects of genotype errors in pedigree studies (e.g. Lincoln & Lander, 1992; Stringham & Boehnke, 1996;
Abecasis et al., 2001), less has been devoted to these issues in studies of population genetic variation. Rice & Holmans (2003) investigated the effect of SNP genotyping errors in the evaluation of a case-control study, demonstrating its importance. They proposed a model to account for error in the calculation of genetic relative risks.

Although tests for deviation from HWE have been used to detect SNPs which contain genotyping errors, as far as I am aware no method has been proposed for using patterns of LD to detect SNP genotype errors in a collection of unphased multilocus genotypes from unrelated individuals. Here I propose such a method based on the models presented at the beginning of this chapter.

The basic idea behind the approach is that genotype errors will result in deviations from expected patterns of linkage disequilibrium. By modelling the errors, LD among the true genotypes \( x \) may still be captured, and the error rates \( \epsilon \) will be estimated to have non-zero values. Ideally one would like to identify every individual genotype error in a data set. However, this seems challenging based solely on patterns of LD. A potentially easier problem is to identify sites at which there exists an elevated rate of genotyping error, and I focus on this problem here.

I assume dense unphased multilocus genotype data \( g \), collected at \( M \) SNP markers in \( n \) individuals. I consider each of the three error models described above, allowing \( \epsilon \) to vary by SNP marker, so that \( \epsilon = (\epsilon_1, \ldots, \epsilon_M) \), where \( \epsilon_m (m = 1, \ldots, M) \) may itself be a vector of rates. The method is motivated by attempting to construct a likelihood ratio (LR) for each SNP \( m \), and then taking the natural log of this LR:

\[
\lambda_m = \log \left( \frac{p(g|\hat{\nu}, \hat{\epsilon})}{p(g|\hat{\nu}, \hat{\epsilon}_{(-m)}\epsilon_m = 0)} \right),
\]

where \( \nu = (\alpha, \theta, r) \), \( (\hat{\nu}, \hat{\epsilon}) \) are the maximum likelihood estimators (MLEs) for \( (\nu, \epsilon) \) under the unconstrained model and \( (\hat{\nu}, \hat{\epsilon}_{(-m)}) \) are the MLEs under the constraint \( \epsilon_m = 0 \), with

\[
\epsilon_{(-m)} := (\epsilon_1, \ldots, \epsilon_{m-1}, \epsilon_{m+1}, \ldots, \epsilon_M).
\]

In practice it is difficult to obtain the MLEs. The likelihood surface will typically
have many modes and convergence of the EM algorithm for parameter estimation may take a large number of iterations. Further, the motivation for maximum likelihood is dubious in this setting with such a highly parameterized model. However, it nonetheless provides motivation for the following algorithm:

1. Estimate $(\hat{\nu}, \hat{\epsilon})$ by fitting the unconstrained model with 25 iterations of the EM algorithm after a random initiation for $(\nu)$ as in section 2.4 and setting $\epsilon_m = .01 \ (m = 1, \ldots, M)$.

2. Set $\nu_m$ to $\hat{\nu}_m$ and Obtain estimates for $\hat{\nu}_m$ from 1 additional iteration of the EM algorithm, constraining $(\nu_{(-m)}, \epsilon_{(-m)})$ to be $(\hat{\nu}_{(-m)}, \hat{\epsilon}_{(-m)})$ and $\epsilon_m$ to be zero.

3. Calculate $\lambda_m$ using (6.1) and the values for $(\hat{\nu}, \hat{\epsilon})$ and $(\hat{\nu}, \hat{\epsilon})$ obtained from the above steps.

4. (Optional) Repeat this process $T-1$ times and combine $\lambda_m^{(1)}, \ldots, \lambda_m^{(T)}$ as follows:

$$
\lambda_m = \log \left( \sum_{t=1}^{T} \exp \lambda_m^{(t)} \right) - \log T.
$$

By using $(\hat{\nu}, \hat{\epsilon}_{(-m)})$ as initial values for $(\nu, \epsilon)$ in step 2, it is hoped that the parameters will correspond to the same mode of $\mathcal{L}(\nu, \epsilon; g)$ obtained with $(\hat{\nu}, \hat{\epsilon})$. Since convergence will not have been achieved with such a small number of iterations, I only estimate $\nu_m = (\alpha_m, \theta_m, r_m)$ in the second step. Otherwise, even small changes in the parameters at other sites will result in $\lambda_m$ being less than one. Presumably, at sites with evidence for larger values of $\epsilon_m$, $\lambda_m$ will be significantly larger than one, providing evidence for elevated rates of errors.

An alternative criterion for detecting genotyping errors is the expected number of genotyping errors at a site $m$, which I calculate as:

$$
\sum_{i=1}^{n} p(x_{im} \neq g_{im} | g, \hat{\nu}, \hat{\epsilon}). \quad (6.2)
$$
This criterion has the advantage that the model may be fit once for all SNPs, with the expected number of errors calculated for each SNP conditional on the estimated parameters. It thus requires less computational effort than the above algorithm (it is on the order of \( M \) times faster).

### 6.4 A Comparison of Error Models

#### 6.4.1 Materials

Large population genetic data sets with known genotyping errors are not generally available. Therefore, it is somewhat of a challenge to test these methods for error detection on data from real populations, where a subset of the genotypes have been observed with some known type of error. However, the HapMap database includes genotype data which did not pass quality control (QC) measures, and these data provide some opportunity for such a test.

Data from the HapMap project consist of filtered and unfiltered data. This filtering involves removing SNPs from the data based on multiple single SNP QC criteria, including the following: low passrate (less than 80% of genotypes were not called); failure to pass concordance tests at a site (duplicate samples were inconsistent); significant deviation from HWE; and, for the CEPH (CEU) and Yoruba (YRI) analysis panels, existence of Mendelian inconsistencies among related individuals. The 60 unrelated individuals from the CEU and YRI samples are the parents from 30 independent parent-offspring trios in each panel, and genotypes from these offspring provide information on the accuracy of the parental genotypes. Since mutations are rare in a single generation, Mendelian inconsistencies may indicate errors in the genotypes of either the parents or offspring. They may also indicate deletions, since current high-throughput genotyping technologies do not explicitly allow for deletions as part of the genotype. The HapMap QC measures dictate that SNPs which exhibit more than one Mendelian inconsistency (MI) are not included in the filtered list.
The *unfiltered* list of SNPs therefore contains SNP sites which were considered to have an elevated probability of genotype errors in the samples. From this unfiltered list of SNPs, I eliminated those which were filtered based on discordance, deviations from HWE and low passrate, leaving only those that passed QC and those with at least 2 MIs. For each of these remaining SNPs, I counted the actual number of MIs by comparing genotypes among individuals in each trio, as the database only designates a SNP as having fewer than 2 MIs (passed QC) or at least 2 (failed due to MIs). A greater number of MIs at a SNP provides stronger evidence of a true error among the unrelated individuals, and sites that contain only one error, and thus perhaps one MI, may show a signal for error although they passed the HapMap QC criterion for MIs.

### 6.4.2 Model Assessment

From the HapMap database (release # 20, January 2006, NCBI build 35), I obtained genotypes from unrelated CEU individuals at 10,000 consecutive SNPs in a 10 Mb region of chromosome 22q. Approximately 1.2% of SNPs contained at least 2 MIs and 1.9% contained exactly 1 MI.

To conduct an initial assessment of the three error models on these data, I compared the distribution of λ for three types of SNPs: those with 0, 1, and at least two MIs. I selected all 124 SNPs with at least two, 150 with exactly one, and 650 with zero MIs. For each model, I computed λ statistics for these 774 SNPs using the data from all 10,000 consecutive SNPs and plotted the empirical CDFs of these statistics in Figure 6.1.

None of the models show a potential to completely discern sites with multiple MIs from those with zero. The median values of λ for sites with multiple MIs are close to zero. At about the 70-th quantile the distributions of λ for the 3 SNP classes begin to differ. (Those for 1-MI and multiple-MI sites are right-skewed.) This suggests that, in practice, it will be difficult to detect more than 30% of the sites with errors.
Figure 6.1: Empirical CDF of $\lambda$. Empirical distributions of $\lambda$ for SNPs with 0, 1, and at least 2 MIs, labeled with “0”, “1”, and “2”, respectively. Values for $\lambda$ are obtained from 5 random starts of EM, with $K = 5$. Sample sizes for each type of SNP are: 124 with at least 2 MIs, 150 SNPs with 1 MI, and 650 with 0 MIs. Often, when $\epsilon_m$ is close to 0, setting $\epsilon_m$ to zero does not provide a real constraint and the small number of additional EM iterations will cause the value of $\lambda_m$ to be $\leq 0$. Only nonnegative values are plotted.
Further, achieving that proportion will incur a large number of false positives, since
the relative frequency of sites with multiple MIs is quite low, around 2%. However,
the distributions do differ among the three types of sites. Among the three error
models, the allelic error model appears to achieve the greatest differentiation of the
distributions of λ among the 3 SNP classes.

Next, I applied all three models to all 10,000 SNPs in an attempt to detect the
sites with MIs. In a scan of a large data set, it is inconvenient to compute λ for
every site and some sort of filter will need to be applied. To reduce computation
time, I computed λ values for SNPs at which there were at least .5 expected genotype
errors, according to (6.2). To assess the ability of each model to discern sites with
multiple MIs using λ as a criterion, I tabulated the number of such sites discovered
(“true positives”) and the number of sites with zero MIs (“false positives”), using
various values of λ as a cutoff value. Note that this criterion does not consider the
sites with exactly one MI. This is justified in part since the sites with 1 MI are likely
an inhomogeneous set: of the genotypes from unrelated individuals, there could very
well be zero errors or multiple errors, given the presence of only one MI. These results
are in Figure 6.2.

Based on the evidence in Figures 6.1 and 6.2, the allelic error model seems slightly
superior at discerning sites with multiple MIs from those with zero (or one). Therefore,
I decided to use this model in the following application to data from the HapMap.

6.5 Evaluation and Validation of Methods for Error Identification

6.5.1 Evaluation of Error-identification Criteria on Unfiltered CEPH Data

Here I apply the method to a different genomic region, chromosome 7q. First, I con-
sider 60,000 SNPs from the unfiltered CEU data of HapMap data release of January,
2006, and NCBI build 35, where I can compare any inferred signatures of genotyping
error with the number MIs observed at each SNP. Using these data, I compare λ
Figure 6.2: *True positives vs. false positives for 3 error models.* True positives are sites with multiple MIs and false positives are sites with zero MIs. Values of $\lambda$ used to classify sites were obtained from 5 random starts of the EM algorithm with $K = 5$, calculated at all sites for which the expected number of genotype errors was at least one-half.
and the expected number of genotyping errors as criteria for discerning SNPs with multiple Mls.

I applied the allelic error model to CEU genotypes from 60,000 SNPs from chromosome 7. As for the chromosome 22 data described above, these SNPs consist of those which failed QC due to the presence of Mls among the related individuals, as well as those which passed QC. In addition, these data include 23 SNPs that contained genotypes from unrelated individuals discordant with genotypes obtained at other genotyping centers. The inclusion of these sites provides an additional opportunity to assess the ability of the method to detect confirmed errors.

In an effort to assess the two criteria for discerning SNPs with errors, $\lambda$ and the expected number of genotyping errors (6.2), I compared the number of true positives with the number of false positives for various values of these criteria. From the results in Figure 6.3, it seems that $\lambda$ may provide a greater ability to discern among sites with zero and multiple Mls at lower levels of FDR (the bottom figure).

Next, in Figure 6.4, I plotted the expected number of genotype errors and values of $\lambda$ calculated from a scan of the 60,000 SNPs. Numerical summaries of these results for the $\lambda$ criterion are in Table 6.1. The right side of the plot contains a set of SNPs that are highly enriched for those with multiple Mls. For example, among the 80 SNPs with $\lambda$ values exceeding 5, 64 had multiple Mls and only 3 were sites with 0 Mls. Of the 23 “discordant” SNPs, 6 of them have $\lambda$ values exceeding 2 with at least 6 expected genotype errors.

As indicated by the horizontal cluster of 1-MI SNPs in Figure 6.4 between 1 and 2 expected genotype errors, this expected-errors criterion may be approximately calibrated for a certain range of the number of errors. Sites with some evidence for elevated error rates (SNPs with $\lambda$ values exceeding 2.7) seem to provide fairly accurate measures of this quantity (Figure 6.5). This criterion may be beneficial if applied in combination with $\lambda$ to improve the method’s ability to differentiate error-prone and error-free sites.
Figure 6.3: *True positives vs. false positives for chromosome 7.* Bottom plot corresponds to a zoomed-in region of the top plot. True and false positives were counted as for Figure 6.2, using $\lambda$ and the expected number of errors as the classification criteria. True positives are sites with multiple MIs. False positives are sites with zero MIs. The values of $\lambda$ used for classification were obtained from the 5 runs of the EM which produced the results in Figure 6.4. By only calculating $\lambda$ at sites with at least .5 expected errors, the number of sites considered was reduced to 5722.
Figure 6.4: Expected number of errors and \( \lambda \). The values are obtained from 5 starts of the EM algorithm. During each run of the EM, values of \( \lambda \) were calculated at SNPs which had at least 0.5 expected errors. \( K \) was set at 5. The \( \circ \), +, and \( \diamond \) symbols represent SNPs with 0, 1, and at least 2 MIs, respectively. The * symbol denotes SNPs with genotypes discordant from other genotypes which did pass QC. The point circled in blue corresponds to the SNP genotypes in Figure 6.6.
Figure 6.5: *Calibration of expected error*. For each value of observed MI, distributions of expected number of errors in SNPs which had exactly this number of MIs are plotted. Only SNPs with significant signal for error ($\lambda > 2.7$) are included in the calculations of expected number of errors.
Table 6.1: *Distribution of SNP error types by \( \lambda \) for CEU chromosome 7.* For values of \( \lambda \) (natural log) from 1 to 5, the number of SNPs with \( \lambda \) values exceeding \( \lambda^* \) are given for SNPs with at least 2, at least 1, and 0 MIs. The "\% of all SNPs with \( \geq 2 \) MIs" column refers to the proportion detected of all multiple-MI SNPs in the collection of 60,000 SNPs from chromosome 7. FDR is calculated as the number of sites with 0 MIs divided by the number of sites with either 0 or at least 2 MIs, i.e., sites with only one MI are excluded from this quantity.

<table>
<thead>
<tr>
<th>( \lambda^* )</th>
<th>No. SNPs &gt; ( \lambda^* ) with:</th>
<th>% of all SNPs with ( \geq 2 ) MIs</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \geq 2 ) MIs</td>
<td>( \geq 1 ) MI</td>
<td>0 MIs</td>
</tr>
<tr>
<td>5</td>
<td>64</td>
<td>77</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>95</td>
<td>155</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>180</td>
<td>299</td>
<td>74</td>
</tr>
<tr>
<td>2</td>
<td>278</td>
<td>470</td>
<td>198</td>
</tr>
<tr>
<td>1</td>
<td>414</td>
<td>726</td>
<td>584</td>
</tr>
</tbody>
</table>

6.5.2 Validation of Methods using Genotype Intensities

To confirm that the method is identifying SNPs which actually contain some sort of genotyping error (including deletions or copy number polymorphisms; CNP), I attempted to gain access to scatter plots of the genotype intensities from which genotype calls were made. I was able to examine 3 such plots from the HapMap (kindly provided by S. McCarroll of the International HapMap Consortium); unfortunately, only one of these corresponded to a SNP which the method flagged as suspicious in terms of containing errors.

The scatter plot for this suspicious SNP is in Figure 6.6. It corresponds to the right-most SNP in the cluster of 3 black circles above 10 expected errors in Figure 6.4, circled in blue. The intensity plot indicates a possible CNP or miscalled genotypes for one of the 3 visible clusters. Since a large number of SNPs are potentially affected by either explanation, it is not surprising the expected number of errors is rather high.
In addition to the above validation, I obtained genotype data from 192 unrelated individuals which were recruited as part of the Pharmacogenetics and Risk of Cardiovascular Disease study (PARC). These data consist of approximately 105,000 SNP markers across the autosomal chromosomes. I analyzed these data (independently by chromosome), calculating $\lambda$ values for any SNPs which demonstrated a total expected number of errors exceeding 5 or for which a single individual contributed at least .8 expected errors. I inspected the genotype intensity data for SNPs which showed strong evidence of genotyping error to further validate the method for error identification.

Among SNPs with the 20-highest values of $\lambda$, 3 lacked evidence of any genotyping error whatsoever. Three-quarters contained obvious deviations from normal genotype intensities; many of these seemed to be CNPs. Genotype intensities of one of the SNPs with errors (along with an example of normal intensities) are in Figure 6.7. Two of the 20 were among a large number of SNPs which showed a fairly strong signal for error but for which the contribution to the total number of errors (among all 192 individuals) was coming mainly from one individual. The consistency with which this individual contributed the majority of the error signal is indicative of that person being of a different demographic background (see discussion in Remarks below), and this situation is fairly easy to identify. Often at these SNPs, all observed genotypes were homozygous for the same allele, except for this individual, who was homozygous for the other allele.

Excluding SNPs which received an error signal from this same individual, among those which I felt confident in classifying either as definitely containing errors or error-free, 82% contained errors. It would be helpful to conduct a more complete survey of these PARC genotype intensities, especially with an expert who could verify my categorization of intensity plots into those which show evidence for CNPs, those likely to contain one or a few errors, and those which look error-free.
Figure 6.6: *Genotype intensities of HapMap data.* The axes are genotype intensities from the Illumina genotyping platform for the two alleles. **LEFT. Example of a normal intensity plot from the HapMap.** In the left plot, the 3 genotype classes are easily visible as clusters of genotypes. The middle cluster at approximately 45 degrees from the origin contains the heterozygote genotypes. **RIGHT. Example of intensities which confirm a SNP suspected to contain genotype errors from the HapMap.** Here, the three clusters may be an indication of increasing copies of a second allele (that is, they all have one common allele corresponding to the vertical position); or they perhaps should be called as homozygotes, heterozygotes, and homozygotes (minor). Instead the two right-most clusters were grouped together and called as heterozygotes.
Figure 6.7: Genotype intensities of PARC data. The vertical axes are for genotype intensities from the Illumina genotyping platform. Horizontal axes display the angle of the points in corresponding plots with “Cartesian” coordinates (these are a transformation of the types of plots in Figure 6.6). LEFT. Example of a normal intensity plot from the PARC study. The 3 genotype classes cluster into 3 groups. RIGHT. Example of intensities which confirm a SNP suspected to contain genotype errors in the PARC data. This plot contains three clusters; however, the “middle” cluster contains a mix of red and pink colored genotypes, indicating that the red genotypes in that cluster were miscalled as homozygotes instead of heterozygotes.
6.6 Identification of Error-prone SNPs in the HapMap Database

Here I apply the method to genotypes from Chinese (CHB) and Japanese (JPT) analysis panels in the non-redundant (filtered) directory of the HapMap database in an attempt to spot suspicious SNPs which might contain genotyping errors. For these CHB and JPT analysis panels of the HapMap project, individuals were not sampled in trios, but instead genotypes of 89 unrelated individuals were obtained between both groups. Because of this design, the CHB and JPT data were not “cleaned” based on the MI criteria that were applied to the CEU and YRI samples.

I applied my method to 105,000 CHB and JPT genotypes of chromosome 7q, using the distribution obtained from application to the CEU data as a guide for assessing significance of the results. I obtained 105,000 SNPs from the combined CHB/JPT sample of the same release and NCBI build as was used for the analysis of CEU data in Section 6.5.1. To decrease computation time, I broke the data into 21 sets of 5,000 consecutive SNPs. The entire analysis took approximately 51 hours on a single 3 GHz Xeon processor with 4 GB of memory.

For this analysis, I do not know which SNPs contained errors and there is no way to detect MIs in the absence of additional data from related individuals. A plot like that of Figure 6.4 for these CHB and JPT data reveals a distribution of criteria with fewer extremely large values than were observed from application to the CEU data at 60,000 SNPs. This is probably because SNPs which were suspected of containing errors in the CEU or YRI samples, via assessment of MIs, were removed from the CHB and JPT databases, since errors might be due to SNP-specific issues.

Still, among these 105,000 SNPs there were 83 which had values of λ exceeding 5. Among the 80 SNPs exceeding 5 in the CEU analysis, only 3 had zero MIs, and 64 had at least 2 MIs. It may be likely that a large proportion of these 80 SNPs do, in fact, contain errors. Based on the presence of 3.9 million SNPs in the filtered, nonredundant database for these CHB and JPT panels, there are approximately 3,050
which would be expected to have \( \lambda \) values that exceed 5, based on the analysis of these 105,000 SNPs.

The use of the observed statistics from the CEU data is more appropriate if the CEPH and Asian panels demonstrate comparable patterns of LD such that values of \( \lambda \) between the two samples are meaningful for comparison. In light of these issues, any extrapolation from one sample to the other should be interpreted with caution.

### 6.7 Remarks

Detecting the majority of genotyping errors based solely on patterns of LD is somewhat impractical. Performance of the method presented here will depend on several factors, including the accuracy of the models for LD and genotyping error. Selection of appropriate models for error detection will depend on the generating mechanism of the errors, the amount of data, the level of LD among loci, as well as the loss function for detecting errors of a particular type. For example, the allelic error model is probably not well-suited for detecting errors resulting from a homozygote of one allele being observed as a homozygote of the other allele. From this perspective, perhaps a combination of models would provide a more comprehensive approach to scanning population genetic data for errors. Additional data sets with different rates, density and mechanisms of genotyping errors would be required for a more complete assessment of these models. Alternatively, simulations may provide some insight into the effects of error model misspecification.

One assumption of the model is that individuals are sampled from a homogeneous population; that is, it is assumed that haplotypes are in HWE. Although results from Chapter 3 indicate that the model is somewhat robust to certain kinds of deviations from HWE for the task of inferring missing genotypes, the methods for error identification are sensitive to the case where one individual from a different population is included in the sample. To establish this, I investigated the effect of adding data from one African American (AA) individual to the sample of 23 individuals of European
descent (ED) from the SeattleSNPs data. These data are of a very high quality, and analysis of only the 23 ED individuals results in two SNPs with $\lambda$ exceeding 5, four exceeding 4, and eight exceeding 3. Inclusion of a single AA individual results in a shift of smaller to moderate $\lambda$ values, with 55 SNPs exceeding 3; however, there were still only four exceeding 4 and two exceeding 5. It is easy to calculate which individual is making the maximum contribution to the error signal (in terms of total expected errors) at every SNP. From this, the contribution from the AA individual is apparent. The data from such an individual could be removed from an analysis to detect genotyping error. (It is worth noting that this is not necessarily the best way to identify individuals from a different population.) If there are at least 2 individuals from a different population and these samples can be identified, then modelling the known (or inferred) population structure using the methods of Chapter 5 eliminates most of the SNPs with inflated signal.

There are several issues which make the application to the HapMap data considered here particularly difficult. First, the absence of MIs does not indicate an error-free site. That is, many SNPs which passed QC may actually contain multiple errors. Additionally, the presence of MIs does not necessarily imply the existence of errors in the genotypes of unrelated individuals. The offspring may contain all of the errors which led to the MIs at a particular site. This will be more likely at sites with fewer MIs (since this requires the offspring to contain all of a smaller number of errors), and thus discerning sites which had only one MI from those which had none is particularly challenging. Because of these issues, a significant number of misclassifications are to be expected.

Since the FDR calculations in Table 6.1 were based only on sites with 0 or at least 2 MIs in the CEU data, these values might be optimistic when applied to data including SNPs with only one error, as these will tend to be the more difficult sites to discern. However, presumably many of the sites without any Mendelian inconsistencies in the CEU sample actually do contain errors, thus inflating the estimated FDR
values. Ideally, an experienced technician would evaluate the original genotyping intensity data for a subset of SNPs which surpassed threshold values of λ to assess the calibration of FDR values.

The pass-rate and HWE QC measures could be performed on population genetic data. Indeed, it would be interesting to see if my method would be able to spot sites flagged by these criteria. However, the check for error-prone sites based on the presence of MIs would not be available without availability of family members and additionally would require a substantial increase in genotyping resources. It would potentially be beneficial if this method could provide an additional filter for genotypes of unrelated individuals.
Chapter 7

CONCLUSION

In this chapter, I present a brief review of Chapters 2–6 and summarize the contributions of this thesis. I review the software developed, which implements methods presented in previous chapters. Finally, I conclude with a discussion of partially-completed and future work which did not receive mention in previous chapters.

7.1 Contributions of this Thesis

In Chapter 2, I presented a model for variation among unphased multilocus genotypes from unrelated individuals, randomly sampled from a population. The model aims to capture patterns of LD among linked loci. Its motivation comes from the observation that over short distances, haplotypes sampled from a population tend to cluster into a few groups of similar haplotypes. I use a Markov chain to model the cluster memberships and allow these to change continuously along a chromosome.

This cluster-based model is less computationally complex than some other models, upon which state-of-the-art methods for haplotype and genotype estimation are based. Because of this computational convenience, the models developed here can be applied directly to unphased genotype data from tens of thousands of individuals, typed at hundreds of thousands of SNPs. The largest data set to which I have successfully applied the model at this time consisted of 290,639 SNPs from 12,000 individuals, a data set resulting from multiple concatenations of data from a Phase-II release of HapMap chromosome 2. Fitting the model to these data would require approximately 259 hours on a single AMD Opteron 875 processor running at 2193 MHz with 16 GB of memory. Methods which can effectively handle large amounts of data are increasingly
necessary with the availability of large-scale genotyping technologies.

The method for haplotype estimation is almost as accurate as the best state-of-the-art method but requires a fraction of the computational cost. For missing genotype estimation the results are as good or better than any other method I have examined. I tested and compared my methods to several of the available methods in Chapter 3.

In Chapter 4, I presented a procedure for approximating the distribution of genotypes at untyped SNPs, which has potential implications in fine-scale LD-based mapping of an association between genotype and phenotype. However, this may have other uses as well, in any situation where this distribution is of interest.

Chapter 5 contains an extension of the model to deal with a certain type of population structure. I demonstrate that this extension improves performance of haplotype inference and missing data applications in the presence of this structure in the situation where it is known. It is also natural to model the LD in an attempt to estimate this structure, using the basic framework developed in this chapter, and this effort has begun to be explored (not by me but building on this work).

Finally, in Chapter 6 I presented a model which accounts for several types of genotyping error. I use this model to develop methodology to detect such errors based on patterns of LD. Such methods may be of use in population studies with large amounts of dense SNP data, such as with case-control genetic association studies.

7.2 Software

Most of the models and methods presented in this dissertation are implemented into a software package called fastPHASE. It is available from the Stephens Software website:

http://www.stat.washington.edu/stephens/software.html

The documentation for fastPHASE is given in Appendix E and covers the methods presented in Chapters 2, 3, 5 and the constant α and r estimators of Chapter 4.
The algorithms necessary for error-modelling and detection (Chapter 6) are fully implemented but not documented at this time. The smoothing methods of section 4.1.2, as well as the procedure involving sampling from columns of $\theta$ (section 4.2), are not fully implemented.

Although the applications presented in this thesis are to unphased multilocus genotype data, the methods of Chapter 2 are first presented in terms of observed haplotypes. The software implementation handles observed haplotype data, as well, so that if some haplotypes are sampled from the population, this information can be utilized. This should improve model fitting, and indeed does provide an increase in performance over sampling an equivalent number of haplotypes observed as unphased genotypes (results not presented). The genotype error model is not implemented for haplotype data.

7.3 Future Work

The methods presented for missing data imputation in Chapters 3 and 4 (section 4.2) are intended to present a framework for assessing genotype-phenotype associations using data collected from unrelated individuals, either from a prospective cohort or a retrospective case-control analysis. The basic idea is to use all available data, including dense SNP data (e.g. from the HapMap), to help inform the distribution of genotypes at untyped loci among a large number of individuals. These individuals might be sampled for the explicit purpose of investigating the genetic causes or associations with certain phenotypes, such as diseases.

The model for genotype errors presented in Chapter 6 models the true latent genotype, possibly observed with some type of error. In developing algorithms for estimating parameters of the model for patterns of LD in the presence of errors (Chapter 2), the framework has been introduced with which I could model other types of observed data, where an underlying model for LD among latent genotypes may be appropriate of helpful. In particular, raw genotype-intensity data are typically processed where
the data are evaluated independently at different sites. Incorporating a model for LD may help in correctly assigning genotypes, in SNP discovery where unphased genotypes are obtained via complete resequencing, or when one does not wish to condition on called genotypes in a 2-step procedure.

Finally, applications of the models for patterns of LD presented here have been to SNP markers only. There is nothing inherent in the development of the markers which necessitates only two allelic types. Recent studies have demonstrated widespread copy number polymorphism (CNP) in the human genome (Conrad et al., 2006; McCarroll et al., 2006) that may span multiple markers. It has been suggested by others (Hinds et al., 2006) that these CNPs may exhibit considerable LD with each other and with existing SNPs. I would like to investigate the utility of incorporating CNP features into this model for LD, potentially using the model to aid in their detection.
BIBLIOGRAPHY


Appendix A

FORWARD AND BACKWARD ALGORITHMS

Two fundamental quantities associated with HMM computations are the so-called “forwards” and “backwards” probabilities:

\[
\phi^m_{\nu}(m, \{k_1, k_2\}) := p(g_{1m}, \ldots, g_{m}, z_{1m} = \{k_1, k_2\}, \nu) \left( \frac{1}{2} \right)^{I(k_1 \neq k_2)},
\]

\[
\beta^m_{\nu}(m, \{k_1, k_2\}) := p(g_{m+1}, \ldots, g_{M}, z^*_{1m} = \{k_1, k_2\}, \nu),
\]

where \( I(A) \) is one if \( A \) is true and zero otherwise. (The factor \( \left( \frac{1}{2} \right)^{I(k_1 \neq k_2)} \) is not usually included in the definition of \( \phi \), but I include it here for later notational convenience.) Although computation of these quantities for HMMs via recursive formulae is standard, I give these formulae here as some care is needed to make these computations \( \mathcal{O}(K^2) \), rather than \( \mathcal{O}(K^4) \).

The forwards calculation is given by

\[
\phi^m_{\nu}(m + 1, \{k_1, k_2\}) = p(g_{(m+1)}|z^*_{(m+1)} = \{k_1, k_2\}, \nu) \left[ p(j_{im} = 0|\nu) \phi^m_{\nu}(m, \{k_1, k_2\}) \right]
\]

\[
+ \frac{p(j_{im} = 1|\nu)}{2} \left( \alpha_{k_1(m+1)} \sum_{k'=1}^{K} \phi^m_{\nu}(m, \{k', k_2\}) + \alpha_{k_2(m+1)} \sum_{k'=1}^{K} \phi^m_{\nu}(m, \{k', k_1\}) \right)
\]

\[
+ p(j_{im} = 2|\nu) \alpha_{k_1(m+1)} \alpha_{k_2(m+1)} \sum_{k_{1}'=1}^{K} \sum_{k_{2}'}^{K} \phi^m_{\nu}(m, \{k_{1}', k_{2}'\})\right],
\]

for \( m = 1, \ldots, M - 1 \), where \( \phi^m_{\nu}(1, \{k_1, k_2\}) := p(g_1|z^*_{1} = \{k_1, k_2\}) \alpha_{k_1} \alpha_{k_2} \). Then \( p(g_{i}|\nu) \) may be calculated as \( \sum_{k_{1}=1}^{K} \sum_{k_{2}=1}^{K} \phi^m_{\nu}(M, \{k_1, k_2\})\).
The corresponding backwards recursion is

\[
\beta^i_v(m - 1, \{k'_1, k'_2\}) = p(j_{im} = 0|\nu)p(g_{im}|z^*_{im} = \{k'_1, k'_2\}, \nu)\beta^i_v(m, \{k'_1, k'_2\}) \\
+ \frac{p(j_{im} = 1|\nu)}{2} \left( \sum_{k=1}^{K} p(g_{im}|z^*_{im} = \{k'_1, k\}, \nu)\beta^i_v(m, \{k'_1, k\})\alpha_{km} \\
+ \sum_{k=1}^{K} p(g_{im}|z^*_{im} = \{k, k'_2\}, \nu)\beta^i_v(m, \{k, k'_2\})\alpha_{km} \right) \\
+ p(j_{im} = 2|\nu) \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(g_{im}|z^*_{im} = \{k_1, k_2\})\beta^i_v(m, \{k_1, k_2\})\alpha_{k_1m}\alpha_{k_2m},
\]

for \(m = 2, \ldots, M\), and with \(\beta^i_v(M, \{k_1, k_2\}) := 1\), for all \(k_1, k_2\).

Finally, I obtain

\[
p(z^*_{im} = \{k_1, k_2\}|g_{i}, \nu) \propto \phi^i_v(m, \{k_1, k_2\})2^{I(k_1 \neq k_2)}\beta^i_v(m, \{k_1, k_2\}),
\]

with the constraint that \(\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(z^*_{im} = \{k_1, k_2\}|g_{i}, \nu) = 1\).
Appendix B

SAMPLING DIPLOTYPES CONDITIONAL ON OBSERVED GENOTYPES

Recall that $d_i$ denotes the pair of haplotypes $(h_i^0, h_i^1)$ for individual $i$. Additionally let $w_i$ denote the ordered pair of cluster-of-origin indicators that correspond to the haplotypes $(h_i^0, h_i^1)$. Thus, $d_i$ and $w_i$ may be thought of as “phased versions” of $g_i$ and $z_i$, respectively. To sample from $p(d_i|g_i, \nu)$, perform the following.

1. Sample $\hat{z}_i$ from $p(\hat{z}_i|g_i, \nu)$. This involves sampling the hidden state $z_i$, conditional on the data $g_i$ and parameters $\nu$, which is a standard procedure for HMMs. First, sample $\hat{z}_{iM} \sim p(\hat{z}_{iM}|g_i, \nu) \propto p(g_i, \hat{z}_{iM}, \nu)$. Then recursively, for $m = M - 1, \ldots, 1$, sample $\hat{z}_{im}$ from:

\[
p(\hat{z}_{im}|\hat{z}_{i(m+1)}, g_i, \nu) \propto p(g_i, \ldots, g_{im}, \hat{z}_{im}, \nu)p(\hat{z}_{i(M+1)}|\hat{z}_{im}, \nu) = \phi^1(m, \hat{z}_{im})G(m, \hat{z}_{im})p_{m+1}(\hat{z}_{im} \to \hat{z}_{i(m+1)}),
\]

where $p_{m+1}(\hat{z}_{im} \to \hat{z}_{i(m+1)})$ is given by (2.10).

2. Sample $\hat{w}_i$ from $p(w_i|g_i, \hat{z}_i, \nu) = p(w_i|\hat{z}_i, \nu)$. Since

\[
p(w_i|\hat{z}_i, \nu) = p(w_i|\hat{z}_{i1}, \nu)p(w_i|\hat{z}_{i2}, \nu) \cdots p(w_i|w_{iM}|w_{i(M-1)}, \hat{z}_{iM}, \nu),
\]

each $\hat{w}_{im}$ may be sampled sequentially (for $m = 2, \ldots, M$), given $\hat{w}_{i(m-1)}$ and $\hat{z}_{im}$. Given $\hat{z}_{im} = \{k_1, k_2\}$, there are at most 2 possibilities for $w_{im}$: $(k_1, k_2)$ and
\((k_2, k_1)\). Thus, probabilities of these outcomes are

\[
p(w_{im} = (k_1, k_2) | \tilde{w}_{i(m-1)} = (k_1', k_2'), z_{im}^* = \{k_1, k_2\}, \nu) \propto p_m(k_1' \rightarrow k_1)p_m(k_2' \rightarrow k_2),
\]

\[
p(w_{im} = (k_2, k_1) | \tilde{w}_{i(m-1)} = (k_1', k_2'), z_{im}^* = \{k_1, k_2\}, \nu) \propto p_m(k_1' \rightarrow k_2)p_m(k_2' \rightarrow k_1).
\]

3. Sample \(\tilde{d}_i\) from \(p(d_i | \tilde{w}_i, g_i, \theta) = \prod_{m=1}^{M} p(d_{im} | \tilde{w}_{im}, \theta)\). This is nontrivial only for heterozygous sites, i.e. when \(g_{im} = 1\). Then,

\[
p(d_{im} = (h_{im}^a, h_{im}^b) | (\tilde{w}_{im}^a, \tilde{w}_{im}^b) = (k_1, k_2), g_{im}, \theta) \propto \theta_{k_1m}^{h_{im}^a}(1 - \theta_{k_1m})^{1-h_{im}^a} \theta_{k_2m}^{h_{im}^b}(1 - \theta_{k_2m})^{1-h_{im}^b},
\]

for \((h_{im}^a, h_{im}^b) = (0, 1), (1, 0)\).
Appendix C

EM ALGORITHM

Here I describe an EM algorithm for the estimation of \( \nu = (\alpha, \theta, r) \). To do this I introduce latent variables, relating to “jumps” that occur in the continuous Markov jump process underlying \( z_i^* \) (Sections 2.2-2.3). Specifically, let \( j_{im} \) denote the number of jumps between markers \( m - 1 \) and \( m \) for individual \( i \), and let \( j_{imk} \) denote the number of these that jump to cluster \( k \). Thus, \( j_{im} := \sum_{k=1}^{K} j_{imk} \), and

\[
p(j_{im} = a | r) = \begin{cases} 
e^{-2r_m d_m}, & a = 0, \\
2 (1 - e^{-r_m d_m}) e^{-r_m d_m}, & a = 1, \\
(1 - e^{-r_m d_m})^2, & a = 2. \end{cases}
\]

Now let \( Q(\nu | \nu^c) \) be the expected complete-data loglikelihood, 

\[
E_{\nu^*} \left[ \log p(g, z, j | \nu) | g \right],
\]

where

\[
p(g, z, j | \alpha, \theta, r) = \prod_{i=1}^{n} p(g_i, z_i, j_i | \alpha, \theta, r),
\]

and

\[
p(g_i, z_i, j_i | \alpha, \theta, r) = p(g_i | z_i, \theta) p(z_i | j_i) p(j_i | \alpha, r).
\]

The algorithm is first initiated with a random guess \( \nu^{(0)} \). Then the following is repeated for \( c = 1, \ldots, C \):

\[
\nu^{(c+1)} = \arg \max_{\nu} Q(\nu | \nu^{(c)}),
\]

for sufficiently large \( C \). The maximization above is accomplished by finding solutions to \( \frac{\partial Q(\nu)}{\partial \theta_{km}} = 0 \), \( \frac{\partial Q(\nu)}{\partial \alpha_{km}} = 0 \), and \( \frac{\partial Q(\nu)}{\partial r_m} = 0 \), for all \( k = 1, \ldots, K \) and \( m = 1, \ldots, M \).
\( r_1 \) is not defined. This leads to the following estimators for \( \nu \):

\[
\hat{\alpha}_{km} = \frac{\sum_i \sum_{k'} \sum_{k'} E_{\nu} [j_{imk} | \theta_{km}^*] \sum_{k'} E_{\nu} [j_{imk}^* | \theta_{km}^*]}{\sum_i \sum_{k'} E_{\nu} [j_{imk} | \theta_{km}^*]}, \tag{C.1}
\]

\[
\hat{r}_m = \frac{-\log \left( 1 - \frac{\sum_i \sum_{k'} E_{\nu} [j_{imk} | \theta_{km}^*]}{2n} \right)}{d_m}, \tag{C.2}
\]

\[
\hat{\theta}_{km} = \frac{A}{B} \tag{C.3}
\]

where,

\[
A = \sum_i \sum_{k'} I_{\{g_{im} \neq 0\}} \left( \frac{\theta_{km}^* (1 - \theta_{km}^*)}{\theta_{km}^* (1 - \theta_{km}^* + \theta_{km}^* (1 - \theta_{km}^*))} \right) I_{\{g_{im} = 1\}} \times p(z_{im} = \{k, k'\} | g_i, \nu^*) 2^{I_{\{\nu = \nu^*\}}}, \text{ and}
\]

\[
B = \sum_i \sum_{k'} p(z_{im} = \{k, k'\} | g_i, \nu^*) 2^{I_{\{\nu = \nu^*\}}}.
\]

In practice, it would be inefficient to calculate \( \hat{r} \) using (C.2) only to exponentiate it as in (2.4). In fact, when using the model of Section 2.2 where \( r_m \) is estimated separately in each marker interval, \((1 - e^{-r_m g_m})\) could be replaced by a single parameter and estimated with \( \sum_i \sum_{k'} E_{\nu} [j_{imk} | \theta_{km}^*] \). However, writing it as above (C.2) facilitates construction of EM algorithms for the constrained model where all \( r_m \) are equal.

Finally, I give expressions for terms necessary in calculating (C.3 - C.2) above. Note that \( p(z_{im} | g_i, \nu) \) depends only on the data for individual \( i \):

\[
p(z_{im} = \{k_1, k_2\} | g_i, \nu) \propto \phi_i^k(m, \{k_1, k_2\}) 2^{k_1 \neq k_2} \beta_i^k(m, \{k_1, k_2\}),
\]

with \( \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(z_{im} = \{k_1, k_2\} | g_i, \nu) = 1 \).

To calculate \( E_{\nu} [j_{imk} | g_i] \) I use \( \sum_{j=0}^{2} j \times p(j_{imk} = j | g_i, \nu) \), which reduces to

\[
E_{\nu} [j_{imk} | g_i] = \frac{\alpha_{km}}{p(g_i, \nu)} \sum_{k=1}^{K} p(j_{im} = 1 | r) \sum_{k' = 1}^{K} \phi_{r}(m - 1, \{k', k''\}) + 2p(j_{im} = 2 | r) p(g_{i(\leq m - 1)} | \nu) \alpha_{km} \right) p(g_{im} | z_{im} = \{k, k'\}, \theta) \beta_{r}(m, \{k, k'\}).
\]
Appendix D

GENOTYPE ERROR ALGORITHMS

Here I give the details of an EM algorithm which jointly estimates $(\alpha, \theta, r, \epsilon)$ for the simple error model of Chapter 6. The basic outline for this EM algorithm is the same as for the previous one in Appendix C. It involves iteratively calculating expected values for latent variables, given a current value of the parameters, and then updating the parameter estimates based on the expectations computed previously. I first derive the forward and backward algorithms for the model with genotyping error. These are necessary for subsequent calculations.

D.1 Forward and Backward Algorithms

Similar to the expressions of Appendix A, $\phi^i_{\nu c}$ and $\beta^i_{\nu c}$ are probabilities:

$$\phi^i_{\nu c}(m, \{k_1, k_2\}) := p(g_{i1}, \ldots, g_{im}, z^*_m = \{k_1, k_2\} | \nu, \epsilon) \left(\frac{1}{2}\right)^{I_{(k_1, k_2)}},$$

$$\beta^i_{\nu c}(m, \{k_1, k_2\}) := p(g_{i(m+1)}, \ldots, g_{iM} | z^*_m = \{k_1, k_2\}, \nu, \epsilon),$$

where $\nu = (\alpha, \theta, r)$. The forwards calculation is given by

$$\phi^i_{\nu c}(m + 1, \{k_1, k_2\}) = p(g_{i(m+1)} | z^*_{i(m+1)} = \{k_1, k_2\}, \nu, \epsilon) \left[p(j_{im} = 0|\nu)\phi^i_{\nu c}(m, \{k_1, k_2\})ight.$$ 

$$+ \frac{p(j_{im} = 1|\nu)}{2} \left(\alpha_{k_1(m+1)} \sum_{k'_{1} = 1}^{K} \phi^i_{\nu c}(m, \{k', k_2\})ight)$$

$$+ \alpha_{k_2(m+1)} \sum_{k'_{1} = 1}^{K} \phi^i_{\nu c}(m, \{k', k_1\})\right)$$

$$+ p(j_{im} = 2|\nu)\alpha_{k_1(m+1)}\alpha_{k_2(m+1)} \sum_{k'_{1} = 1}^{K} \sum_{k'_{2} = 1}^{K} \phi^i_{\nu c}(m, \{k', k_2\})\right].$$
for \( m = 1, \ldots, M - 1 \), where \( \phi_{\nu \epsilon}^i(1, \{ k_1, k_2 \}) \) := \( p(g_{i1}|z_{i1} = \{ k_1, k_2 \}, \nu, \epsilon)\alpha_{k_11}\alpha_{k_21} \) and

\[
p(g_{im}|z_{im} = \{ k_1, k_2 \}, \theta, \epsilon) = \sum_{a=0}^2 p(g_{im}|x_{im} = a, \epsilon)p(x_{im} = a|z_{im} = \{ k_1, k_2 \}, \theta).
\]

(D.1)

The expression \( p(g_{im}|x_{im}, \epsilon) \) is given by the relevant error model (Section 6.2). The second factor in the summand of (D.1) is given by (2.12) for \( x_{im} = 0, 1, 2 \). Then the probability of the data for one individual may be calculated as

\[
p(g_{i}|\nu, \epsilon) = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \phi_{\nu \epsilon}^i(M, \{ k_1, k_2 \}),
\]

for \( i = 1, \ldots, n \).

The backwards recursion is

\[
\beta_{\nu \epsilon}^i(m - 1, \{ k'_1, k'_2 \}) = p(j_{im} = 0|\nu)p(g_{im}|z_{im} = \{ k'_1, k'_2 \}, \nu, \epsilon)\beta_{\nu \epsilon}^i(m, \{ k'_1, k'_2 \})
\]

\[
+ \frac{p(j_{im} = 1|\nu)}{2} \left( \sum_{k=1}^{K} p(g_{im}|z_{im} = \{ k', k \}, \nu, \epsilon)\beta_{\nu \epsilon}^i(m, \{ k'_1, k \})\alpha_{km} \right)
\]

\[
+ \sum_{k=1}^{K} p(g_{im}|z_{im} = \{ k, k'_2 \}, \nu, \epsilon)\beta_{\nu \epsilon}^i(m, \{ k, k'_2 \})\alpha_{km}
\]

\[
+ p(j_{im} = 2|\nu) \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(g_{im}|z_{im} = \{ k_1, k_2 \}, \nu, \epsilon)
\]

\[
\times \beta_{\nu \epsilon}^i(m, \{ k_1, k_2 \})\alpha_{k_1m}\alpha_{k_2m},
\]

for \( m = 2, \ldots, M \), and with \( \beta_{\nu \epsilon}^i(M, \{ k_1, k_2 \}) := 1 \), for all \( k_1, k_2 \). For the parameter estimators, I require the following:

\[
p(z_{im}^* = \{ k_1, k_2 \}|g_i, \nu, \epsilon) \propto \phi_{\nu \epsilon}^i(m, \{ k_1, k_2 \})2^{I(k_1\neq k_2)}\beta_{\nu \epsilon}^i(m, \{ k_1, k_2 \}),
\]

(D.2)

with the constraint that \( \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(z_{im}^* = \{ k_1, k_2 \}|g_i, \nu, \epsilon) = 1 \).

**D.2 EM Algorithm**

Here I present estimators for \( \alpha, \theta, r, \) and \( \epsilon \), based on the simple error model, allowing error rates to vary by SNP. The basic idea is the same for the 2-parameter simple
and allelic models, with an extra parameter at each site since there are two error
rates for these. To facilitate the construction of the algorithm, let $e_{im}$ denote an error
event, equaling one if an event occurred and zero otherwise. Conditional on an error
event the observed genotype is selected uniformly at random from all three possible
genotypes. That is, for all $i$ and $m$,

$$p(e_{im}|e_m) = (e_m)^{e_{im}}(1-e_m)^(1-e_{im}),$$

for $0 \leq e \leq 1$, and

$$p(g_{im}|e_m = 1) = \frac{1}{3},$$

for $g_{im} = 0, 1, 2$.

Let $Q(\nu, \epsilon|\nu^*, \epsilon^*)$ denote $E_{\nu^*, \epsilon^*} \log \left[p(g, x, z, j, \epsilon|\alpha, \theta, r, \epsilon)|g\right]$, the expected
complete-data loglikelihood, where the complete-data likelihood is given by

$$p(g, x, z, j, \epsilon|\alpha, \theta, r, \epsilon) = \prod_{i=1}^{n} p(g_i, x_i, z_i, j_i, e_i|\alpha, \theta, r, \epsilon),$$

with

$$p(g_i, x_i, z_i, j_i, e_i|\alpha, \theta, r, \epsilon) = p(g_i|e_i, x_i, \epsilon)p(x_i|z_i, \theta)p(z_i|j_i)p(j_i|\alpha, r).$$

Setting the partial derivatives of $Q$ to zero and solving for $(\alpha, \theta, r, \epsilon)$ leads to the
following estimators:

$$\hat{\alpha}_{km} = \frac{\sum_i E_{\nu^*, \epsilon^*} [j_{imk}|g]}{\sum_i \sum_k E_{\nu^*, \epsilon^*} [j_{imk}|g]},$$

$$\hat{\theta}_m = -\log \left(1 - \frac{\sum_k E_{\nu^*, \epsilon^*} [j_{imk}|g]}{2n}\right),$$

$$\hat{e}_m = \frac{\sum_i p(e_{im} = 1|g, \nu^*, \epsilon^*)}{n},$$

$$\hat{\theta}_{km} = \frac{A}{B}.$$
where

\[ A = \sum_i \sum_{k'} \tilde{p}(z_{im} = \{k, k'\}|g, \nu^*, \epsilon^*) 2^{\tilde{r}_{(k'=k)}} \left[ \frac{\theta_{km} (1 - \theta_{km})}{\theta_{km} (1 - \theta_{km}) + \theta_{km} (1 - \theta_{km})} \right] \times p(x_{im} = 1|g, \nu^*, \epsilon^*) + p(x_{im} = 2|g, \nu^*, \epsilon^*) \], and

\[ B = \sum_i \sum_{k'} \tilde{p}(z_{im} = \{k, k'\}|g, \nu^*, \epsilon^*) 2^{\tilde{r}_{(k'=k)}}. \]

These expressions are valid assuming all data are observed. If there are missing genotypes, the estimators must be appropriately modified. For example, in \( \hat{c}_m \), only observed genotypes are used and the denominator may be replaced by a count of observed genotypes. The estimators for \( \alpha, \theta \), and \( r \) are similar to those given in Appendix C, with probability expressions (including expectations) now involving on \( \epsilon \).

Estimators for \( \alpha \) and \( r \) depend on the expected number of jumps in an interval to particular clusters \( (j_{ink}) \) and in total \( (\sum_k j_{ink}) \). These expectations are given by

\[
E_{\nu, \epsilon}[j_{ink}|g] = \frac{\alpha_{km}}{p(g_k \mid \nu, \epsilon)} \left[ \sum_{k'=1}^K p(j_{im} = 1|g) \sum_{k''=1}^K \phi_{i, k''}(m - 1, \{k', k''\}) + 2p(j_{im} = 2|g)p(g_{i(=m-1)} \mid \nu, \epsilon) \alpha_{km} \right] \times p(g_{im} \mid z_{im} = \{k, k'\}, \theta, \epsilon) \beta_{i, k''}(m, \{k, k'\}),
\]

for \( m = 1, \ldots, M \) and \( k = 1, \ldots, K \).

Finally I give a formula for computing the expected values of \( \epsilon \), given the observed data. If \( g_{im} \neq x_{im} \) then \( e_{im} \) is clearly one. However, an error event need not lead to an actual error. This leads to the following expression:

\[
p(e_{im} = 1|g, \nu, \epsilon) = \frac{2}{n} \sum_{a=0}^{2} (q)_{I(a=a_{im})} \times p(x_{im} = a|g, \nu, \epsilon),
\]
where $q = \frac{\xi}{3 + (1 - e)}$, the probability that an error event occurred, given $g_m = \hat{x}_m$, and

$$p(x_m | g_{i}, \nu, \epsilon) \propto p(g_m | x_m, \epsilon) \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(x_m | z_m = \{k_1, k_2\}, \theta) \times \left[ \phi_{i\epsilon}^i(m - 1, \{k_1, k_2\}) p(j_m = 0 | r) + \left( \alpha_k m \sum_{k'=1}^{K} \phi_{i\epsilon}^i(m - 1, \{k', k_2\}) \right) \frac{p(j_m = 1 | r)}{2} \right. \left. + \alpha_{k_1 m} \sum_{k'=1}^{K} \phi_{i\epsilon}^i(m - 1, \{k', k_1\}) \right] \times \beta_{i\epsilon}^i(m, \{k_1, k_2\}),$$

with the constraint $\sum_{a=0}^{2} p(x_m = a | \cdot) = 1$. As with the algorithm of Appendix C, this EM algorithm is $O(K^2 Mn)$, linear in the amount of data and $K^2$. 
Appendix E

DOCUMENTATION FOR FASTPHASE 1.1

Introduction

fastPHASE is software that implements methods for estimating missing genotypes and
reconstructing haplotypes from unphased SNP genotype data of unrelated individuals.
The methods are based on a cluster model for haplotypes, which is described in Scheet
& Stephens (2006). Parameters of the model are first estimated with an EM algorithm;
then conditional on these parameters, missing genotypes and haplotypes are inferred.

E.0.1 What is new in this version?

- The default choice for the number of clusters is now determined by fastPHASE
  (previously default was 10).

- The `-s` option allows for sampling haplotypes conditional on observed geno-
types.

- `-U` allows for simulation of haplotypes from the model underlying fastPHASE,
  unconditional on observed data.

- `-F` provides estimates of sample haplotype frequencies.
Getting started

E.0.2  Installation

Presumably, you have downloaded a fastphase.VERSION.linux.tar.gz file. (Alternatively, you downloaded a version for Darwin\(^1\), the Mac OS X flavor of UNIX, or for Microsoft Windows\(^2\), in which case substitute for “linux” below.) Then do the following:

\[
\text{gunzip fastphase.VERSION.linux.tar.gz}\\
\text{tar \ -xvf fastphase.VERSION.linux.tar}
\]

This will produce a folder similar in name to fastphase.\text{VERSION}.linux.

Containing in this new directory are the following:

- fastPHASE

- fastphase.inp  an example input file

- fastphase.haplotypes.inp  an example input file of known haplotypes

- fastphase.subpopslabels.inp  file of “subpopulation” labels for individuals in fastphase.inp

- fastphase.subpopslabels2.inp  contains enough labels to run with \text{-b \fastphase.subpopslabels2.inp} flags.

To do a short run and test that the program is working on your system:

\[
./\text{fastPHASE \ -T1}
\]

\(^1\)To use fastPHASE on a Mac, you’ll need to open a “terminal” session by launching the Terminal application. This is typically in Applications → Utilities.

\(^2\)To use fastPHASE on a PC running Microsoft Windows, launch a MS-DOS window by going to Start → Run and type “cmd”. You may have to find another way to unzip and untar the compressed file you downloaded.
E.0.3 Usage

You may be reminded of some available options from the command line by running

```
./fastPHASE -h
```

which will give the current usage.

Please note that **there should be no spaces between an option flag and any number or filename** that follows. In the following (for example)

```
./fastPHASE -T10 -ulabels.txt -oMyresults test.inp
```

I'm specifying that I'd like only 10 starts/runs of the EM algorithm (instead of 20). Additionally, here I've chosen to use *Myresults* as a prefix for the main output files that fastPHASE creates. Finally, I'm supplying a file *labels.txt* which contains subpopulation labels for the individuals in my input file *test.inp*.

**Input file format**

The input filename may be supplied by the user after any options; otherwise, fastPHASE looks for a file in the current directory named *fastphase.inp*. Information required in the input file includes: the number of diploid individuals to be analyzed, the number of SNP sites in the data, and the genotypes for each individual. Optionally, the file may specify a label for each individual (in fact, this is assumed). Additionally, the file may contain the relative physical positions of the SNP markers and may contain a line of 'S' characters. The physical positions may be utilized in conjunction with a different model option. The S-line is allowed only for compatibility with PHASE input files, as fastPHASE only works with SNP data in its current implementation.

An individual’s unphased genotypes are to be provided on 2 lines using any characters other than the question mark (?), with ? denoting missing alleles. Spaces and tabs may separate the allele characters. The input file may be represented as follows:
no.individuals
no.SNPsites
P pos(1) pos(2) ... pos(no.SNPsites) <optional line>
SSS...SSS <optional line>
ID (1)
genotypes(1-a)
genotypes(1-b)
ID (2)
genotypes(2-a)
genotypes(2-b)
.
.
.
ID (no.individuals)
genotypes(no.individuals-a)
genotypes(no.individuals-b)

where the quantities are the following:

- **no.individuals** An integer specifying the number of individuals who have been genotyped.

- **no.SNPsites** An integer specifying the number of SNP sites at which each individual has been typed.

- **P** The character *P* (upper case).

- **pos(i)** A number indicating the position of site *i*. The sites must be in their physical order along the chromosome (i.e., these positions must be increasing). With the default settings, this information is ignored, and so this “P-line” may be omitted.
• SSS... A line of S characters (optional).

• ID(i) A character string, i.e. label, for individual i. If you do not wish to specify a label for each individual, omit them and use the -n option.

• genotypes(i-a) and genotypes(i-b) Genotypes for individual i.

For example, consider this small example input file, test.inp:

```
3
4
# id 1
1a11
0t01
# id 2
1t11
0a00
# id 3
?a01
?t10
```

A few comments:

• In the above example there are 3 individuals typed at 4 SNP sites. The first individual is heterozygous at 3 sites and homozygous for 1 at the last site, the second individual is heterozygous at all 4 sites, and the third individual contains missing data at the first site.

• Above, the second SNP is coded with a/t, but for each SNP site, any 2 characters may be used for the two SNP alleles.

• When giving the genotypes, you may place spaces between adjacent alleles, e.g.
would be acceptable for the last line.

- fastPHASE does not explicitly or comprehensively check the format of the input file; therefore, it is recommended that you verify that the output (in the \_genotypes\.out or \_hapgues\.switch\.out files) is consistent with your input data.

**Output**

Output files for inferred haplotypes or imputed genotypes contain two lines per given diploid individual, with the order of individuals corresponding to that supplied in the input file. In addition, summary information is given, such as a recapitulation of some of the parameters from the fastPHASE run, and the command line supplied by the user. The \_switch\.out file contains estimates which attempt to minimize the switch error, and the \_indiv\.out file contains estimates which attempt to minimize individual error (see Stephens & Donnelly, 2003, for a review of these error measures). If haplotypes are not estimated (by supplying a negative integer following \\_h), fastPHASE creates a \_genotypes\.out file, containing the original data and estimated unphased genotypes which were denoted as missing in the input file.

**Available options**

You may be reminded of available options by supplying the \_h option.

**E.0.4 Input & output options (\_n, \_o)**

- \_n id lines omitted in input file. The input file consists of only 2 lines per diploid individual (in addition to the "header" information such as the number of individuals, number of SNPs, etc.).
• `-o` specifying output file prefix. For example, `-oMyresults` would result in the output file: `Myresults_hapguess_switch.out`

### E.0.5 Controlling the algorithm (`-T`, `-C`, `-H`, `-i`)

- `-T<number>` number of random starts of the EM algorithm. The default is 20, although 10 starts produced results close to those from 20 in our tests (in half the time, of course).

- `-C<number>` the no. of iterations of the EM algorithm. For our tests, the default of 25 was sufficient. However, you may see a small gain in accuracy from more iterations on larger datasets (I have only limited data on this). This value has worked well for data from 60 unrelated individuals typed at 42,000 SNPs.

- `-H<number>` set the number of haplotypes sampled from the “posterior” distribution obtained from a particular random start of the EM algorithm. Since the default number of EM runs is 20, `-H50` would produce 1000 samples of consecutive 2-site haplotypes, which I have observed to be sufficient for minimizing switch error. If you wish to minimize individual error (less relevant for datasets with many SNPs), you may want to set this value higher (e.g., 200). If your only interest is inferring missing genotypes, you can turn off haplotype estimation with `-H-4` (or some negative integer), to save computation time.

- `-i` estimate haplotypes by minimizing individual error. If you specify the `-i` option, an additional file is printed for estimated haplotypes: `fastphase_hapguess_indiv.out`, where “fastphase” may be replaced with text supplied after the `-o` flag.
E.0.6  *Determination of number of clusters (–K[...])*

The default action for fastPHASE is to determine the number of haplotype clusters via a cross-validation procedure (see Scheet & Stephens, 2006). **This default is new in version 1.1 and for smaller data sets** (containing at most 500 SNP loci) **this add considerably to the running time!** To turn this off and select the default from previous versions (which was 10), use –K10. **Use of the –K option followed by an integer** takes precedence over the –K options below which control the cross-validation algorithm. The number of clusters selected by the procedure is given in the output file (unless the –Z option is given to simplify the output). The default range of 5, 10, 15 may not include large enough values for complex data sets, with large numbers of individuals, or for data with large amounts of haplotypic diversity. The user should be aware of this and set appropriate limits using the controls below.

The cross-validation procedure consists of searching over a range of values for the number of clusters $K$. To accomplish this, fastPHASE applies missingness (masks data) to a portion of the observed data and, for several values of $K$, makes a best-guess for the missing genotypes. This process is repeated multiple times, each time choosing a different portion of the observed data (all individuals but only up to a certain number of SNP loci), and the value of $K$ is chosen which produced the lowest overall error rate.

- **–Ku<number>** or **–KU<number>**  *upper limit for no. of clusters.* Largest value of $K$ considered during cross-validation.

- **–KL<number>** or **–KL<number>**  *lower limit for no. of clusters.* Smallest value of $K$ considered during cross-validation.

- **–Ki<number>** or **–KI<number>**  *interval between values for number of clusters.* This controls the difference between values of $K$ considered for cross-validation.
For example, if the lower and upper limits for $K$ are 5 and 14, respectively, the 
$\texttt{-Ki3}$ option would require fastPHASE to consider values 5, 8, 11, and 14.

- $\texttt{-Ks<number> or -KS<number>}$  \textit{no. of times masking process applied.} Default\textsuperscript{3} is 10.

- $\texttt{-Km<number> or -KM<number>}$ \textit{no. of SNP loci used for cross-validation.} Default is 500. If data consist of fewer than 500 sites, all will be used.

- $\texttt{-Kp<float> or -KP<float>}$ \textit{rate at which missingness (masking) applied.} Default is .1.

An example usage for these options is:

\texttt{-KL6 -KU12 -Ki2 -Ks50 -Km1000 -Kp.05}

which would tell fastPHASE to do the following 50 times: randomly select at most 1000 consecutive SNPs; mask approximately 5% of the observed genotypes among all individuals at 1000 SNPs; consider 6, 8, 10, and 12 for the number of clusters; impute missing genotypes at each value of $K$ and tabulate errors.

Results from this $K$-selection process are written to a file with a \texttt{kselect} extension. This file includes the parameters from the procedure, as well as a list of values of $K$ considered and the corresponding number of genotypes imputation errors (and error rate).

\textbf{E.0.7 Utilizing known haplotypes ($\texttt{-b, -B}$)}

Known haplotypes may be supplied in a separate file if the $\texttt{-b<filename>}$ flag is specified. \textit{Only entire haplotypes may be specified – this is NOT equivalent to the $\texttt{-k}\textsuperscript{3}With this and all defaults, use the \texttt{-h} (help) option to verify default values. Default values printed in this documentation may not reflect actual values implemented in fastPHASE.}
flag in PHASE. The input file for known haplotypes should be as follows:

no.haplotypes
ID hap 1
haplotypes(1)
ID hap 2
haplotypes(2)
.
.
.

haplotypes(no.haplotypes)

where the above quantities are the following:

- no.haplotypes Integer specifying the number of haplotypes.

- haplotypes(i) Alleles for haplotype i, consisting of no.SNP sites characters that correspond to the same characters used in the main input file.

Please note that this file does not contain its own line for the number of SNP sites. This is assumed to be the same as for the main input file for unphased genotypes.

If all of the genetic data is of haplotypes (i.e. all of data is phase-known) and you want to estimate parameters and simulate or estimate missing alleles, you may notify fastPHASE that your main data file contains haplotype data with the -B flag. The haplotypes may be listed one after the other (without “id labels”) in conjunction with the -n option, or grouped into “pseudo-individuals” with one “id line” separating pairs of haplotypes.

E.0.8 Incorporation of subpopulation labels (-u)

Subpopulation labels may be supplied with the -u<filename> flag. The information must be supplied on one line with different integers for different labels. (All values
okay except -9999.) Here are example contents of such a file, which would correspond to the test.inp file above.

5 2 5

This specifies that the first and third individuals were sampled from the same subpopulation. If known haplotypes are supplied, fastPHASE expects additional labels, one for each haplotype. So if labels are provided for the analysis of 3 diploid genotypes and 2 haplotypes, there should be 5 integers on one line in the labels file.

E.0.9 Sampling and simulating (-s, -U)

- **-s** sample haplotypes given observed unphased genotypes. Instead of producing point estimates of haplotypes or genotypes, you may wish to obtain multiple samples of the haplotypes (and genotypes where missing values were observed), given the observed data. In combination with -H<number>, this option will produce <number> diplotypes (phased haplotypes) for each diploid individual in the sample and <number> haplotypes for each haplotype in the sample for each start of the EM algorithm. These haplotypes are printed to a file with a _sampledWgivG.txt extension. For each sample from the posterior distribution, the haplotypes are given in the same order as the corresponding genotypes of the input file.

- **-U<number>** sample haplotypes unconditional on observed data. The model is fit (for each run of EM) and <number> haplotypes are simulated from the fitted model, unconditional on the observed genotype data.

E.0.10 Estimating haplotype frequencies (-F)

Sample haplotype frequencies may be estimated by Monte Carlo methods with -F. Samples from the posterior distribution of haplotypes given the observed genotypes
are drawn for each run of the EM algorithm. After all runs are completed, the sampled haplotypes are tabulated and frequency estimates are produced. Haplotypes are sampled from the observed genotypes 5,000 times per diploid individual. To increase this number (for more precise estimates), supply a larger number immediately after \(-F\).

Haplotypes with corresponding estimated frequencies are output to a file with the \_freqs suffix. If subpopulation labels are supplied with the \(-u\) option, the \_freqs file will contain separate haplotype frequency estimates for each subpopulation, as well as estimates assuming a single panmictic population.

\subsection*{E.0.11 Options not intended for general use}

- \(\texttt{-Z}\) \textit{print simplified output format}. Produces a simple output, with 2 lines per individual, without “id” lines, subpopulation labels or summary information from the run.

- \(\texttt{-M<number>}\) \textit{modelling options}. The default model is to allow values for \(\alpha\) and \(r\) to vary across the chromosome or genomic region (see Schet & Stephens, 2006). Alternatively, these parameters can be assumed constant across the chromosome. That is, \(\alpha_s = \alpha_t\) and \(r_s = r_t\) (for all \(s, t = 1, \ldots, M\)). To specify a “constant \(\alpha\), constant \(r\)” model, use \(\texttt{-M1}\); for “constant \(\alpha\) only”, use \(\texttt{-M2}\); and for “constant \(r\) only”, use \(\texttt{-M3}\). \textbf{If using models which assume constant values of \(r\) or \(\alpha\), supply the relative physical postions of SNPs via the P line} (see Section E.0.3). During the averaging of results over multiple starts of the EM algorithm, haplotype and genotype inference results may be averaged over different models. To use a “constant \(\alpha\), constant \(r\)” model for half the runs, use \(\texttt{-M41}\); to use “constant \(\alpha\) only” for half the runs, use \(\texttt{-M42}\).
- **-g** turn off genotype imputation. This may save time if you have a large amount of missing data, and you only wish to infer phase (not actually infer the missing genotypes), or some specific simulation tasks.

- **-H<negative-number>** turn off haplotype inference. You can turn off haplotype estimation with -H-4 (or some negative integer), to save computation time, if your only interest is inferring unphased genotypes.

- **-S<number>** set the seed for random number generation. The value of <number> should be a positive integer. If not supplied, seed will be taken from the system clock.

- **-S<negative-number>** read parameter values from file. If a negative integer is supplied, e.g. -S-7, the parameter values are initiated by reading the values from files. These filenames are hardwired to be: alphahat.txt, thetahat.txt, rhat.txt.

- **-p** print estimated parameter values. Can be used to capture parameter values obtained at the end of the final run of the EM algorithm. The filenames produced are the same as above, for use with -S<negative-number>.

**Regarding the analysis of large amounts of data**

The largest data set to which I have fit the model is from The HapMap Project (International HapMap Consortium, 2005): 3150 individuals typed at 290,639 SNPs, which is a result of concatenating 15 times data from chromosome 2.

There are no built-in limits to the amount of data which can be analyzed. The algorithm is linear in both the number of individuals and number of SNPs. Some considerations are
• The maximum number of characters per line which can be read by fastPHASE is currently set to 500,000. So if you have data at a very large number of sites, you may break these up over multiple lines. You should do so in the same fashion as you would with one line, only adding a few breaks where necessary. That is, initially assign phase arbitrarily, giving one "haplotype" on multiple lines and then the other haplotype.

• Since with hundreds of thousands of loci on even a hundred individuals, fitting the model for multiple starts will take perhaps a day or two. To reduce computation time, you might consider reducing the number of starts to 5 or 10, e.g. with -T10.

• Depending on the data, you might consider reducing the value of $K$. We have found -$K8$ to be suitable under a variety of scenarios, although a further reduction may be appropriate for some data. The $K$-selection procedure chooses for the number of clusters, the value which produced the lowest error rate. In practice, these error rates may be very similar, in which case it would be prudent to choose the smallest value of $K$ which still seems reasonable. You may first run the $K$-selection procedure with

\[-C0 -T1 -KL6 -KU14 -Ki2\]

for example. Then, choose $K$ based on the output from this procedure, and set it manually with -$K<number>$.

How to cite this program

In publications which use results from the use of fastPHASE, please cite Scheet & Stephens (2006).
Acknowledgements

The executable for Microsoft Windows was created with DJGPP, which is available from http://www.delorie.com/djgpp/

Obtaining the software

fastPHASE is available for download from:
http://www.stat.washington.edu/stephens/software.html
VITA

Paul Anthony Scheet was born in Portland, Oregon. He spent his youth playing basketball and enjoying life with his sister Margo and his parents, Kenneth and Priscilla. As a teenager, he developed an interest in evolution from studying the fossil records of early hominids.

He graduated from Washington University in Saint Louis in 1995 with a Bachelor of Arts in biology, receiving the Marion Smith Spector Prize for his honors thesis on improving prosthesis selection for total hip replacement surgery. For two years he worked in shotgun assembly and gap finishing in DNA sequencing projects of *C. elegans* and *H. sapiens* at the Genome Sequencing Center at the Washington University School of Medicine.

In 1998, Paul returned to school to improve his background in computational techniques for eventual study of population genetics. At the University of Iowa, he took his first course in statistics and began to develop a passion for the field as it shaped the beginnings of his philosophy of science and knowledge. After receiving his Master of Science in statistics in 2000 from the University of Iowa, he moved to Seattle to study statistical genetics at the University of Washington. In Seattle he made his most important discovery: Larkin Strong, whom he married in 2004. In September 2006, Paul will become a Postdoctoral Research Fellow in the Center for Statistical Genetics at the University of Michigan.