Reconstruction by deconstruction: diplotype frequency estimation for genotype data in stratified populations

Author: Marieke Vinkenoog
s1136305

First supervisor: Dr. Stefan Böhringer
Leiden University Medical Center

Second supervisor: Prof.dr. Aad van der Vaart
Mathematical Institute, Leiden University

MASTER THESIS

Defended on August 29, 2018

Specialisation: data science

STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL SCIENCES

Contents

Chapter 1

Introduction

In the age of big data, many researchers are interested in the big data set that people carry around every day: the human genome. It contains approximately 1.5 Gigabytes worth of information per cell in the haploid genome, and although over 99% is identical for all humans, the remaining part accounts for many individual properties that are of great interest for many research areas.

The entire human genome was first sequenced in 2001. Since then, many genomic studies have made use of the resulting data. One type of study, often used in medical research, is a genome-wide association study (GWAS). GWASs aim to find associations between genetic variants in the DNA that are shared between many individuals, and traits like human diseases. Thousands of relevant (disease) associations have been found in this way, which have played a major role in personalised medicine.

Our DNA is the result of DNA that we inherit from both our biological parents. It therefore makes sense to include pedigree information of DNA in a genetic analysis. However, regular genome testing only provide an individual's DNA, not its origin. Methods to infer this information exist, but they make certain assumptions, such as that populations are in Hardy-Weinberg equilibrium (see section 1.1). This thesis aims to develop methods that relax this assumption, so that the information can be retrieved more accurately.

Section 1.1 of this introduction will summarise and explain some genetic and biological concepts for those that do not have a background in biology, or would like the refresher. Section 1.2 will outline the problem statement and the further structure of the thesis.

1.1 Genetic and biological concepts

DNA

Genetic material in all living organisms is stored in DNA. DNA molecules are long chains of nucleotides, which come in four varieties: adenine (A), thymine (T), guanine (G) and cytosine (C). Sequences of nucleotides can form genes, which are regions of DNA that code for a functional molecule. Due to evolution, mutations can occur within genes, leading to the emergence of several alleles (variations) in a population. It is often the case that two alleles exist for one gene, one being dominant and the other recessive.

A single strand of human DNA is composed of approximately three billion of these nucleotides. DNA has the form of a double helix, two complementary strands of DNA coiled around each other. This long double strand is condensed into several chromosomes, which come in pairs (23 pairs for humans): one inherited from the mother, and one from the father. When both chromosomes carry the same allele for a certain gene, the individual is called homozygous for that gene; when the alleles are different, it is called heterozygous.

SNPs

Single-nucleotide polymorphisms (SNPs) are one of the most common types of genetic variation, and are used in many applications of genome testing. A SNP is a mutation in one single nucleotide of the genome, where all possible alleles (usually just two) have a frequency of at least 1%. Depending on where in the DNA the SNP occurs (the locus), it can fall within coding or non-coding regions of the DNA. In general, SNPs are most likely to be found within non-coding regions, and are therefore unaffected by natural selection.

Considering a single locus with two alleles (a and A), there are three possible genotypes: two homozygous genotypes, $\{A, A\}$ and $\{a, a\}$, and one heterozygous genotype $\{A, a\}$. However, there are two ways in which the heterozygous genotype can occur: the individual could have either inherited the A allele from its mother and the a allele from its father, or vice versa.

As the number of loci increases, so does the number of possible genotypes. At two loci (locus one having alleles a and A, and locus two b and B), four haplotypes exist: $\{ab\}, \{aB\}, \{Ab\}$ and $\{AB\}$. Now, different combinations of haplotypes lead to the same genotype. Genotype $\{AaBb\}$ could arise from either the combination of $\{ab\}$ and $\{AB\}$, or from $\{aB\}$ and $\{Ab\}$. To see the difference between these, the diplotype needs to be known, rather than the genotype. A diplotype consists of two haplotypes, which are the alleles of any number of loci on a single chromosome. In the above example, there would be 16 possible genotypes, leading to only nine unique genotypes (see section 2.4 for the full solution).

Population stratification

When evolutionary influences such as natural selection or genetic drift are absent and new generations are formed through random mating, genotype frequencies remain the same over generations. This principle is called the Hardy-Weinberg equilibrium (HWE). In the simplest case, looking at one bi-allelic locus, only one parameter is needed to describe the haplotype frequencies: $f(a) = \rho$ and $f(A) = 1 - \rho$. This parameter can also describe all genotype frequencies (see section 2.1). HWE can also be viewed as the independence of haplotypes.

In practice, populations are often not in HWE due to random fluctuations, sample bias or population stratification. Population stratification occurs when the population is a mixture of multiple subpopulations that do not interbreed. Over time, this may lead to systematic differences in allele frequencies between subpopulations, and the total population does not adhere to HWE anymore. These differences can pose problems when they occur in population data used in genetic association studies. Such a study might indicate a locus to be associated with a disease, when it is actually due to the structure of the population.

1.2 Problem statement

In genetic (association) studies, data almost always consists of individuals' genotypes, as these are easiest to sequence. When the interest of the study is in haplotypes, these can be retrieved from the data using existing estimation procedures, but the assumption of independent haplotypes is needed, i.e. the population must be in Hardy-Weinberg equilibrium. When this assumption does not hold, the models become unidentifiable because the data does not contain enough information to estimate all parameters.

The aim of this thesis is to obtain several models that do not depend on the assumption of Hardy-Weinberg equilibrium and to develop estimation procedures for the diplotype frequencies. The models will first be applied to simulated data sets, and compared to existing models that do assume the population to be in HWE. Last, data from the HapMap project will be used to illustrate the methods on a real data set.

Chapter 2

Haplotype estimation

2.1 Notation

Consider data on a single locus, with alleles $\{a, A\}$. The genotype score G^s is defined as the number of A alleles in the genotype:

$$G^s = \begin{cases} 0, & G = \{a, a\} \\ 1, & G = \{a, A\} \\ 2, & G = \{A, A\} \end{cases}$$

Haplotypes are denoted by $h_i,\ i \in \{0, ..., 2^M - 1\}$, with M the number of loci that are considered. In this case, there are two haplotypes: $h_0 = (a), h_1 = (A)$. The corresponding haplotype frequencies are written as η_i.

Diplotypes are denoted by $d_{ij},\ i, j \in \{0, ..., 2^M - 1\}$, i being the index of the haplotype inherited by the mother, j that of the haplotype inherited by the father. The corresponding diplotype frequencies are written as δ_{ij}. Diplotypes can be used in two ways: ordered or unordered. Ordered diplotypes takes parental origin into account, unordered diplotypes do not. For instance, for unordered diplotypes, (a, A) and (A, a) would be considered the same, but not for ordered diplotypes. This thesis will work with ordered diplotypes, although symmetry will be assumed (see section 2.2).

For a single locus, there are four diplotypes:

$$\begin{aligned} d_{00} &= (a, a) \\ d_{01} &= (a, A) \\ d_{10} &= (A, a) \\ d_{11} &= (A, A) \end{aligned}$$

Two of these diplotypes, d_{01} and d_{10} result in the same genotype score of 1.

Probabilities of genotypes are denoted by π_p where p indicates the genotype score:

$$\begin{aligned} \pi_0 &:= P(a, a) && = P(G^s = 0) \\ \pi_1 &:= P(a, A) && = P(G^s = 1) \\ \pi_2 &:= P(A, A) && = P(G^s = 2), \end{aligned}$$

where $\sum_{p=0}^{2} \pi_p = 1$.

If the conditions for HWE hold, then the above probabilities can be calculated as follows: $\pi_p = f_p(\rho), p = \{0, 1, 2\}$, where ρ is the frequency of allele a as before.

$$\begin{aligned} f_0(\rho) &= \rho^2 \\ f_1(\rho) &= 2\rho(1-\rho) \\ f_2(\rho) &= (1-\rho)^2, \end{aligned}$$

This can be generalised to more loci by adding one parameter for each locus. π_{pq} is now a tuple of two genotype frequencies, p indicating the genotype score on the first locus, q on the second. Multiplying the frequencies of the alleles that are present in the genotype gives the expected frequency of the genotype. For two loci, $\pi_{pq} = f_{pq}(\rho, \phi), p, q = \{0, 1, 2\}$, where $\rho = P(a)$ for the first locus and $\phi = P(b)$ for the second locus, the frequencies are:

$$\begin{aligned} f_{00}(\rho, \phi) &= \rho^2\phi^2 \\ f_{01}(\rho, \phi) &= \rho^2 * 2\phi(1-\phi) \\ f_{02}(\rho, \phi) &= \rho^2(1-\phi)^2 \\ f_{10}(\rho, \phi) &= 2\rho(1-\rho)\phi^2 \\ f_{11}(\rho, \phi) &= 2\rho(1-\rho) * 2\phi(1-\phi) \\ f_{12}(\rho, \phi) &= 2\rho(1-\rho)(1-\phi)^2 \\ f_{20}(\rho, \phi) &= (1-\rho)^2\phi^2 \\ f_{21}(\rho, \phi) &= (1-\rho)^2 * 2\phi(1-\phi) \\ f_{22}(\rho, \phi) &= (1-\rho)^2(1-\phi)^2 \end{aligned}$$

2.2 Deviation from HWE

Consider two loci. Locus 1 has alleles $\{a, A\}$ and locus 2 $\{b, B\}$. This gives the following possible haplotypes:

$$h_0 = (ab), h_1 = (aB), h_2 = (Ab), h_3 = (AB)$$

These haplotypes can be combined into ordered diplotypes d_{ij}. This gives 16 possible diplotypes with corresponding frequencies. However, since it is assumed that $\delta_{ij} = \delta_{ji}$, only the 10 diplotype frequencies shown in black the table below are needed to describe the parameter space.

	h_0	h_1	h_2	h_3	$\sum$
h_0	δ_{00}	δ_{01}	δ_{02}	δ_{03}	η_0
h_1	δ_{10}	δ_{11}	δ_{12}	δ_{13}	η_1
h_2	δ_{20}	δ_{21}	δ_{22}	δ_{23}	η_2
h_3	δ_{30}	δ_{31}	δ_{32}	δ_{33}	η_3
					1

The sum of all diplotype frequencies must equal one, and summing over a full row (or column) gives the marginalised haplotype frequency.

$$\sum_{i,j} \delta_{ij} = 1 \tag{2.1}$$

$$\sum_{j} \delta_{ij} = \eta_i \tag{2.2}$$

When a population is in Hardy-Weinberg equilibrium, the diplotype frequency is simply the product of the two corresponding haplotype frequencies. When the conditions for HWE do not hold, more parameters are needed to describe the data. The parameter θ_{ij} indicates for each diplotype its frequency relative to its theoretical frequency under HWE conditions. It is defined as follows:

$$\theta_{ij} = \frac{\delta_{ij}}{\eta_i \eta_j} - 1 \tag{2.3}$$

Therefore, a diplotype frequency can be calculated by multiplying the relevant haplotype frequencies and the deviation from HWE.

$$\delta_{ij} = \eta_i \eta_j (1 + \theta_{ij}) \tag{2.4}$$

When $\theta_{ij} = 0$, the diplotype frequency is once again equal to the product of the corresponding haplotype frequencies.

Since $\sum_{i,j} \delta_{ij} = 1$, $10 - 1 = 9$ parameters are needed to describe δ_{ij} and ensure identifiability. Three parameters are used to describe the haplotype frequencies η_i (as they must also sum to one). This leaves six deviation parameters θ_{ij}, which is four fewer than the possible ten (one θ_{ij} for each unique δ_{ij}).

Combining equations (2.2) and (2.4) leads to the following constraint:

$$\begin{aligned}\sum_j \eta_i \eta_j (1 + \theta_{ij}) &= \eta_i \\ \sum_j \eta_j (1 + \theta_{ij}) &= 1 \\ 1 + \sum_j \eta_j \theta_{ij} &= 1 \\ \sum_j \eta_j \theta_{ij} &= 0\end{aligned} \tag{2.5}$$

These are in effect four constraints (one for each i). Therefore, only six θ_{ij} are free, and the others can be calculated from those six. θ_{ij} corresponding to genotypes with at least one heterozygous locus are included in the parameter space. This leads to the following parameter space to describe $\delta_{ij}, i, j \in \{0, 1, 2, 3\}, i \leq j$:

$$\{\eta_0, \eta_1, \eta_2, \theta_{01}, \theta_{02}, \theta_{03}, \theta_{12}, \theta_{13}, \theta_{23}\}$$

Generalizing to M loci, to describe $\delta_{ij}, i, j \in \{0, ..., 2^M - 1\}, i \leq j$, the following parameters are needed:

$$\{\eta_0, ..., \eta_{2^M-1}, \theta_{ij}, i, j \in \{0, ..., 2^M - 1\}, i < j\}$$

2.3 Estimating haplotype frequencies

Consider diplotype data on two loci, where Hardy-Weinberg equilibrium does not hold: $D = (d_1, ..., d_N)$ where the diplotype for an individual n is written as $d_n = (d_{n1}, d_{n2})$.

The haplotype frequencies are the following:

$$\eta_0 := P(ab), \eta_1 := P(aB), \eta_2 := P(Ab), \eta_3 := P(AB), \sum_i \eta_i = 1$$

The complete data likelihood in terms of the diplotype frequencies is:

$$\begin{aligned} \mathcal{L}(D;\delta) &= \prod_{n=1}^{N} \delta_{d_{n1}d_{n2}} \\ &= \prod_{\substack{i=0\\j=0}}^{M} (\delta_{ij})^{N_{ij}}, \end{aligned} \tag{2.6}$$

Where N is the total number of individuals, and N_{ij} the number of individuals with diplotype d_{ij}.

Because symmetry is assumed, the log-likelihood becomes the following:

$$\begin{aligned} \log \mathcal{L}(D;\delta) &= \sum_{i,j}^{M} N_{ij} \log \delta_{ij} \\ &= \sum_{i}^{M} N_{ii} \log \delta_{ii} + \sum_{\substack{i,j\\i<j}}^{M} (N_{ij} + N_{ji}) \log \delta_{ij} \end{aligned} \tag{2.7}$$

The likelihood is subject to the constraint that all diplotype frequencies must sum to 1. This corresponds to the following because symmetry is assumed:

$$\begin{aligned} \sum_{i,j}^{M} \delta_{ij} &= 1 \\ \sum_{i}^{M} \delta_{ii} + 2\sum_{\substack{i,j\\i<j}}^{M} \delta_{ij} &= 1 \end{aligned} \tag{2.8}$$

To maximise the likelihood subject to this constraint, a Lagrange multiplier λ is used and the Lagrange function is optimised instead. The constraint needs to be written as equal to 0, so 1 is subtracted from each side of equation (2.8). The Lagrange function is:

$$\log \mathcal{L}(D;\delta,\lambda) = \sum_{i}^{M} N_{ii} \log \delta_{ii} + \sum_{\substack{i,j\\i<j}}^{M} (N_{ij} + N_{ji}) \log \delta_{ij} - \lambda\Big(\sum_{i}^{M} \delta_{ii} + 2\sum_{\substack{i,j\\i<j}}^{M} \delta_{ij} - 1\Big) \tag{2.9}$$

Taking the partial derivative for each δ_{ij} leads to the following set of equations:

$$\frac{N_{ij} + N_{ji}}{\delta_{ij}} = 2\lambda \quad \text{for } i, j \in \{0, 1, 2, 3\}, i \leq j \tag{2.10}$$

Therefore δ can be expressed in terms of the data and λ as:

$$\hat{\delta}_{ij} = \hat{\delta}_{ji} = \frac{N_{ij} + N_{ji}}{2\lambda} \quad \text{for } i, j \in \{0, 1, 2, 3\}, i \leq j \tag{2.11}$$

Combining equations (2.8) and (2.13) gives the following estimate for λ:

$$\begin{aligned}
\sum_{i}^{M}\frac{N_{ii}}{\lambda}+2\sum_{\substack{i,j\\i<j}}^{M}\frac{N_{ij}+N_{ji}}{2\lambda}&=1\\
\frac{1}{\lambda}\sum_{i}^{M}N_{ii}+\frac{2}{2\lambda}\sum_{\substack{i,j\\i<j}}^{M}(N_{ij}+N_{ji})&=1\\
\frac{1}{\lambda}N&=1\\
\lambda&=N
\end{aligned}\tag{2.12}$$

Now $\hat{\delta}$ can be expressed in terms of the data only by combining equations (2.11) and (2.12):

$$\hat{\delta}_{ij}=\hat{\delta}_{ji}=\frac{N_{ij}+N_{ji}}{2N}\quad\text{for } i,j\in\{0,1,2,3\},i\leq j\tag{2.13}$$

Combining equations (2.2) and (2.13) allows $\hat{\eta}$ to be expressed in terms of the data:

$$\begin{aligned}
\hat{\eta}_i&=\sum_{j}^{M}\hat{\delta}_{ij}\\
&=\sum_{j,i\leq j}^{M}\frac{N_{ij}+N_{ji}}{2N}\\
&=\frac{1}{2N}\sum_{j,i\leq j}^{M}(N_{ij}+N_{ji})
\end{aligned}\tag{2.14}$$

Finally, $\hat{\theta}$ can be calculated by plugging the estimates from equations (2.13) and (2.14) into equation (2.3):

$$\hat{\theta}_{ij}=\frac{\hat{\delta}_{ij}}{\hat{\eta}_i\hat{\eta}_j}-1\tag{2.15}$$

Therefore, when the diplotypes are known, $(\hat{\eta},\hat{\theta})$ can be estimated directly from the data.

2.4 Measured genotypes

When data consists of measured diplotypes, the parameters described in the previous section lead to an identifiable model. However, in practice data are usually measured genotypes, which means that several diplotypes cannot be distinguished. Recall that π_{pq} indicates the frequency of the genotype resulting from genotype score p on the first locus and q on the second, the genotype score being the number of A or B alleles. Possible genotypes and their ordered diplotypes are, for two loci:

$$\begin{aligned}
\pi_{00} := P(aabb) &= P(ab, ab) \\
\pi_{01} := P(aaBb) &= P(ab, aB) + P(aB, ab) \\
\pi_{02} := P(aaBB) &= P(aB, aB) \\
\pi_{10} := P(Aabb) &= P(ab, Ab) + P(Ab, ab) \\
\pi_{11} := P(AaBb) &= P(ab, AB) + P(AB, ab) \\
&\quad + P(Ab, aB) + P(aB, Ab) \\
\pi_{12} := P(AaBB) &= P(aB, AB) + P(AB, aB) \\
\pi_{20} := P(AAbb) &= P(Ab, Ab) \\
\pi_{21} := P(AABb) &= P(Ab, AB) + P(AB, Ab) \\
\pi_{22} := P(AABB) &= P(AB, AB)
\end{aligned}$$

Depending on the observed genotype, three cases for the diplotype can be distinguished:

1. Both loci are homozygous. In this case, the ordered diplotype is unique. This is the case for $\pi_{00}, \pi_{02}, \pi_{20}$ and π_{22}.

2. One locus is heterozygous. Two ordered diplotypes are possible (with swapped parental origins), each with probability $\frac{1}{2}$. The unordered diplotype is still unique. This is the case for $\pi_{01}, \pi_{10}, \pi_{12}$ and π_{21}.

3. Both loci are heterozygous. Two unordered diplotypes are possible, and for each of those, two ordered diplotypes are possible. This is only the case for π_{11}.

Because it is assumed that ordered diplotypes with swapped parental origins are equally likely, the genotype frequencies can be expressed in the diplotype frequencies as follows:

$$\begin{aligned}
\pi_{00} := P(aabb) &= \delta_{00} \\
\pi_{01} := P(aaBb) &= 2\ \delta_{01} \\
\pi_{02} := P(aaBB) &= \delta_{11} \\
\pi_{10} := P(Aabb) &= 2\ \delta_{02} \\
\pi_{11} := P(AaBb) &= 2\ (\delta_{12} + \delta_{03}) \\
\pi_{12} := P(AaBB) &= 2\ \delta_{13} \\
\pi_{20} := P(AAbb) &= \delta_{22} \\
\pi_{21} := P(AABb) &= 2\ \delta_{23} \\
\pi_{22} := P(AABB) &= \delta_{33}
\end{aligned}$$

For all genotypes, the probability of the diplotype is either:

- 0, if the diplotype does not result in the relevant genotype;
- 1, if both loci are homozygous;
- $\frac{1}{2}$, if one locus is heterozygous;
- conditional on the diplotype frequencies, if both loci are heterozygous.

This translates into the following conditional distribution for ordered diplotypes $D = (d_1, ..., d_N)$, where $\Phi(g_i)$ defines the set of diplotypes d that result in genotype g_i.

$$P(d_n|g_i, \eta, \theta) = \left\{ \begin{array}{ll} 0, & d_n \notin \Phi(g_i), \\ 1, & g_i = (0,0), d_n = (0,0) \vee g_i = (2,2), d_n = (3,3) \\ \hline \eta_1\eta_2(1+\theta_{12})/\pi_{g_i}, & g_i = (1,1), d_n \in \{(1,2),(2,1)\} \\ \eta_0\eta_3(1+\theta_{03})/\pi_{g_i}, & g_i = (1,1), d_n \in \{(0,3),(3,0)\} \\ \hline \frac{1}{2}, & else \end{array} \right. \tag{2.16}$$

Where $\pi_{g_i} = \sum_d P(d, g_i|\eta, \theta)$. For $g_i = (1,1)$ this works out to: $\pi_{g_i=(1,1)} = 2\delta_{12} + 2\delta_{03} = 2\eta_1\eta_2(1+\theta_{12}) + 2\eta_0\eta_3(1+\theta_{03})$.

The problem occurs when two or more loci are heterozygous. For two loci, this happens only for π_{11}, which can be the result of either d_{12} or d_{03}. The frequency of this genotype is therefore the sum of the two diplotype frequencies. Now, there are 9 possible genotypes and 9 parameters to describe them, meaning that the model is no longer identifiable and another constraint is needed.

This distribution can be generalised to as many loci as desired. In this thesis, only two or three loci will be analysed at a time. The derivation of the conditional distribution for ordered diplotypes of three loci can be found in Appendix A.

In the next sections, several methods will be proposed to fix this unidentifiability problem.

2.5 Expectation-maximisation algorithm

The diplotypes are considered to be the full data, and are needed for the likelihood as in equation (2.6). The genotypes are the observed data. All N_{ij} can be calculated directly from the observed data, except for N_{03} and N_{12}, which are indistinguishable because the genotypes are the same. These counts are replaced by expectations based on estimates of (η, θ). The maximum likelihood estimates are made using an expectation-maximisation (EM) algorithm. This algorithm was first described in a 1977 paper by Dempsey, Laird and Rubin [2]. It alternates between the expectation (E) step and maximisation (M) step to find the maximum likelihood parameters for the observed data. The EM algorithm works as follows:

1. Choose initial values for $\hat{\eta}$ and $\hat{\theta}$:

$$\begin{aligned} \hat{\eta}_i &= \frac{1}{M} \qquad && \text{for } i \in \{0, ..., M-1\} \\ \hat{\theta}_{ij} &= 0 && \text{for } i, j \in \{0, ..., M-1\} \end{aligned}$$

2. E-step: For each individual, calculate the probability of each diplotype based on the conditional probability defined in equation (2.16), given the data and current values of $(\hat{\eta}, \hat{\theta})$;

3. M-step: based on the expected likelihood of step 2, calculate estimates for $(\hat{\eta}, \hat{\theta})$. The new

estimate of $\hat{\delta}_{ij}$ is the mean probability of this diplotype over all individuals:

$$\begin{aligned}
\hat{\delta}_{ij} &= \frac{\sum_n^N P(d_{n_1,n_2} = \delta_{ij})}{N} \\
\hat{\eta}_i &= \sum_j \hat{\delta}_{ij} \\
\hat{\theta}_{ij} &= \frac{\hat{\delta}_{ij}}{\hat{\eta}_i \hat{\eta}_j} - 1
\end{aligned}$$

4. Repeat steps 2-3 until the differences between the old and new estimates are smaller than some value ϵ.

2.6 Estimation procedure - fixed deviation parameter

A rather basic solution to the unidentifiability problem is to fix one of the deviation parameters θ_{ij}, thus assume it known, and calculate the other parameter based on that value. For two loci, there are two unidentifiable deviation parameters, corresponding to diplotypes δ_{03} and δ_{12}, which both result in the genotype heterozygous on both loci. One can be expressed in terms of the other:

$$\begin{aligned}
\pi_{11} &= 2\eta_1\eta_2(1+\theta_{12}) + 2\eta_0\eta_3(1+\theta_{03}) \\
&= 2\eta_1\eta_2 + 2\eta_1\eta_2\theta_{12} + 2\eta_0\eta_3(1+\theta_{03}) \\
\theta_{12} &= \frac{2\eta_1\eta_2 + 2\eta_0\eta_3(1+\theta_{03}) - \pi_{11}}{-2\eta_1\eta_2} \\
&= \frac{\frac{1}{2}\pi_{11} - \eta_0\eta_3(1+\theta_{03})}{\eta_1\eta_2} - 1
\end{aligned} \tag{2.17}$$

The MLE for the parameters can now be calculated as follows:

1. Choose a value for θ_{03};
2. Estimate $\hat{\eta}$ and $\hat{\theta}$ using the EM algorithm (see section 2.5).

The limits for θ_{ij} can be calculated analytically from the constraints as follows:

$$\begin{array}{rcccl}
0 & < & \eta_i\eta_j(1+\theta_{ij}) & < & 1 \\
0 & < & \eta_i\eta_j + \eta_i\eta_j\theta_{ij} & < & 1 \\
-\eta_i\eta_j & < & \eta_i\eta_j\theta_{ij} & < & 1 \\
-1 & < & \theta_{ij} & < & \frac{1-\eta_i\eta_j}{\eta_i\eta_j}
\end{array} \tag{2.18}$$

Therefore the value chosen for θ_{03} should be between -1 and $\frac{1-\eta_0\eta_3}{\eta_0\eta_3}$. However, since η_0 and η_3 are estimated after the value for θ_{03} is chosen, there should be a check after step 2 to ensure that the values for the parameters are valid.

This algorithm is not very useful on its own, since the researcher needs to know one of the deviation parameters based on prior knowledge. However, the algorithm will prove useful later on, when a way to estimate the deviation parameter is introduced.

2.7 Estimation procedure - diplotype frequency heuristics

In the previous section, an educated guess on one of the deviation parameters was needed in order to estimate the other parameters. The deviation parameters are always unobserved, and are therefore difficult to estimate intuitively or based on prior research. Another way to solve the unidentifiability problem is to put a constraint on the diplotype frequencies, and calculate the deviation parameters from those.

For data on two loci, there are two diplotypes that cannot be distinguished: $\{ab/AB\}$ and $\{aB/Ab\}$. The corresponding diplotype frequencies are δ_{03} and δ_{12}. It might be assumed that individuals with genotype $\{AaBb\}$ are equally likely to have either diplotype, and thus $\delta_{03} = \delta_{12} = \frac{1}{4}\pi_{11}$. When this constraint is enforced, the EM algorithm can uniquely estimate all $(\hat{\eta}, \hat{\theta})$.

Of course, the assumption does not need to be that both diplotype frequencies are equal; it might also be of the form $\delta_{03} = c * \delta_{12}$, where c is a specified constant. This leads to the diplotype frequencies $\delta_{03} = \frac{c\pi_{11}}{2(c+1)}$ and $\delta_{12} = \frac{\pi_{11}}{2(c+1)}$.

To generalise this method to more loci, all sets of diplotypes that return the same genotype should be assigned the same frequency ratio.

2.8 Estimation procedure - stratified populations

A special case of populations in which Hardy-Weinberg equilibrium does not hold are stratified populations. This means that the population is divided into several subpopulations with differing haplotype frequencies. The subpopulations are in HWE, but the total population is not. If two subpopulations are considered, the haplotype frequencies in the total population are:

$$\eta_i = \alpha\eta_i^1 + (1-\alpha)\eta_i^2, \tag{2.19}$$

where α is the mixing rate, and superscripts indicate subpopulations.

Since both subpopulations are in HWE, the diplotype frequencies within the subpopulations are the product of the haplotype frequencies:

$$\delta_{ij}^1 = \eta_i^1\eta_j^1 \qquad \delta_{ij}^2 = \eta_i^2\eta_j^2 \tag{2.20}$$

The diplotype frequencies in the total population are:

$$\begin{aligned} \delta_{ij} &= \alpha\delta_{ij}^1 + (1-\alpha)\delta_{ij}^2 \\ &= \alpha\eta_i^1\eta_j^1 + (1-\alpha)\eta_i^2\eta_j^2 \end{aligned} \tag{2.21}$$

Thus the deviation parameters can be expressed as follows (from equation (2.3)):

$$\theta_{ij} = \frac{\alpha\eta_i^1\eta_j^1 + (1-\alpha)\eta_i^2\eta_j^2}{(\alpha\eta_i^1 + (1-\alpha)\eta_i^2)(\alpha\eta_j^1 + (1-\alpha)\eta_j^2)} - 1 \tag{2.22}$$

When both subpopulations have the same haplotype frequencies, the total population is also in HWE and all θ_{ij} are equal to 0:

$$\begin{aligned} \eta_i^1 &= \eta_i^2 = \eta_i \\ \eta_j^1 &= \eta_j^2 = \eta_j \\ \theta_{ij} &= \frac{\eta_i\eta_j}{\eta_i\eta_j} - 1 = 0 \end{aligned} \tag{2.23}$$

The expectation for mixed populations where subpopulations have different haplotype frequencies is that the actual frequency of homozygous individuals is lower than the expected frequency under HWE. In the most extreme case, subpopulation 1 consists only of individuals with homozygous dominant haplotypes, and subpopulation 2 of individuals with homozygous recessive haplotypes, leading to an observed frequency of 0 for heterozygous haplotypes. The θs corresponding to heterozygous diplotypes will be below 0, and those corresponding to homozygous diplotypes will be above 0.

The inclusion of the parameter for the mixing rate α can be used to accurately identify all θ_{ij} when a reliable estimate of α can be made based on prior research or knowledge. The procedure is then as follows:

1. Set α to the known value;
2. Choose a trivial initial value for $\hat{\theta}_{03}$;
3. Estimate $\hat{\eta}$ and $\hat{\theta}$ using the EM algorithm (see section 2.5);
4. Estimate $\hat{\eta}^1$ and $\hat{\eta}^2$ from $\hat{\eta}$ and α (equation (2.19)), using numerical optimisation;
5. Recalculate $\hat{\theta}_{03}$ using $\hat{\eta}_0^1$, $\hat{\eta}_0^2$, $\hat{\eta}_3^1$, $\hat{\eta}_3^2$ and α (equation (2.22));
6. Repeat 3-5 until the difference between the old and new estimate for $\hat{\theta}_{03}$ is smaller than some value ϵ.

This method has the additional advantage that it will also give the estimates for $\hat{\eta}^1$ and $\hat{\eta}^2$, even without knowing which individuals belong to which subpopulation. It is an improvement over the estimation procedure using a fixed deviation parameter, as accurate estimations of the mixing rate are more often available, and also more meaningful to the researcher.

The numerical optimisation in step 4 works as follows: the expected diplotype frequencies in the total population are given (constant). Different values of haplotype frequencies in the subpopulations are tried, which are used to calculate the diplotype frequencies in the subpopulations, and then using the mixing rate, the diplotype frequencies in the total population. The values are optimised so that the difference between the given diplotype frequencies and the estimated diplotype frequencies are minimal.

Chapter 3

Simulations

In this chapter, the models laid out in the previous chapter will be tested on several simulated scenarios. For each simulation, two sets of parameters will be estimated:

1. Null model: using a standard EM algorithm that assumes the population to be in HWE, which only estimates $\hat{\eta}$;
2. Full model: using the EM algorithm from 2, but both $\hat{\eta}$ and $\hat{\theta}$ are estimated. Several methods for estimating $\hat{\theta}$ are implemented, which vary by scenario.

Each scenario is simulated two times: for sample sizes $N = 100$ and $N = 500$. Scenarios for unstratified populations will be repeated for both 2 and 3 loci at a time. Each simulation is repeated 500 times, after which the Root Mean Square Error (rMSE) is reported for the all estimates of $(\hat{\eta}, \hat{\theta})$ mentioned above. The rMSE is defined as:

$$rMSE(\hat{\eta}) = \sqrt{\frac{\sum_{i=0}^{M}(\hat{\eta}_i - \eta_i)^2}{M}} \tag{3.1}$$

rMSE was chosen as error estimator because it is on the same scale as the parameters it is applied to. For instance, if a haplotype frequency is equal to 0.1, and the rMSE for that parameter is also 0.1, this means an error of 100%. As haplotype frequencies are always between 0 and 1, an rMSE of 0.1 indicates in this case that the estimator on average deviates from the true value of the parameter by 10 percentage points in either direction.

3.1 Scenario 1: unstratified populations in HWE

Data is simulated from an unstratified population that is in Hardy-Weinberg equilibrium. The full model is not expected to perform better than the null model. In fact, the full model will be capturing random noise in its $\hat{\theta}$, which may lead to a worse estimation of $\hat{\eta}$.

Simulation procedure

1. Simulate diplotype frequencies:

$$\begin{aligned} &\eta_i \sim \text{Dir}(\beta_i = 1000) \qquad \text{for } i \in \{0, ..., (2^{Nloci} - 1)\} \\ &\delta_{ij} = \eta_i \eta_j \\ &\big(\theta_{ij} = 0\big) \end{aligned}$$

2. Simulate diplotype data for N individuals using diplotype frequencies as probabilities;
3. Convert diplotypes to genotype data.

Estimation procedure full model

1. Choose initial values for $\hat{\eta}$ and $\hat{\theta}$;
2. Calculate the expected likelihood for the data given the current parameter values;
3. Set new values for $\hat{\eta}$ based on the expected likelihood for the data;
4. Set new values for $\hat{\theta}$ using the values for $\hat{\eta}$ and the data:
    a) If the corresponding diplotype is identifiable (at most 1 locus is heterozygous):
    $\hat{\delta}_{ij} = \frac{N_{ij}+N_{ji}}{2N}$
    $\hat{\theta}_{ij} = \frac{\hat{\delta}_{ij}}{\hat{\eta}_i \hat{\eta}_j} - 1$;
    b) Otherwise: $\hat{\theta}_{ij} = 0$.
5. Repeat 2-4 until the parameters no longer change.

Results

The root mean square errors over all iterations are shown in Table 3.1. All models perform well in estimating the haplotype frequencies; the estimated parameter values are, on average, within 3.5 percentage points of the true values. Increasing the sample size five-fold leads to an rMSE decrease of over 50% in all parameter estimates.

As expected, the null model performs well, and the full model does not perform better. The rMSE in the full model is slightly larger than in the null model. It should be noted that although performance-wise any model could be used, the full model is much more computationally expensive and takes much longer to run than the null model.

Table 3.1: Simulation results scenario 1

		Null model		Full model		
# loci	N	rMSE($\hat{\eta}$)	rMSE($\hat{\delta}$)	rMSE($\hat{\eta}$)	rMSE($\hat{\theta}$)	rMSE($\hat{\delta}$)
2	100	.0348 (.0010)	.0114 (.0049)	.0348 (.0010)	.2408 (.0458)	.0170 (.0047)
2	500	.0153 (.0002)	.0050 (.0022)	.0154 (.0002)	.1014 (.0070)	.0075 (.0022)
3	100	.0233 (.0002)	.0044 (.0009)	.0249 (.0003)	.3316 (.0444)	.0066 (.0011)
3	500	.0128 (.0001)	.0022 (.0006)	.0135 (.0001)	.1705 (.0113)	.0034 (.0006)

3.2 Scenario 2: unstratified population not in HWE

For this scenario, the simulated population is still unstratified, but deviates from Hardy-Weinberg equilibrium for other reasons. These reasons could include any form of non-random mating or genetic drift. The full model assumes that genotypes with more than one heterozygous loci are equally distributed over the diplotypes that result in that genotype. For two loci, this means that $\hat{\delta}_{03} = \hat{\delta}_{12} = \frac{1}{2}\pi_{11}$ (see section 2.4).

It is expected that the full model will out-perform the null model, since θ_{ij} are not equal to zero.

Simulation procedure

1. Simulate diplotype frequencies, calculate population parameters:

$$\begin{aligned}
\beta_{ij} &= 0.8 && \text{for } i,j \in \{0,...,2^{Nloci-1}\}, i \neq j \\
&= 1.2 && \text{for } i,j \in \{0,...,2^{Nloci-1}\}, i = j \\
\delta_{ij} &\sim \text{Dir}(\beta_{ij}) && \text{for } i,j \in \{0,...,(2^{Nloci}-1)\} \\
\eta_i &= \sum_j \delta_{ij} \\
\theta_{ij} &= \frac{\delta_{ij}}{\eta_i \eta_j} - 1
\end{aligned}$$

2. Simulate diplotype data for N individuals using diplotype frequencies as probabilities;

3. Convert diplotypes to genotype data.

Estimation procedure full model

1. Choose initial values for $\hat{\eta}$ and $\hat{\theta}$;

2. Calculate the expected likelihood for the data given the current parameter values;

3. Set new values for $\hat{\eta}$ based on the expected likelihood for the data;

4. Set new values for $\hat{\theta}$ using the values for $\hat{\eta}$ and the data:

   a) If the corresponding diplotype is identifiable (at most 1 locus is heterozygous):
   $\hat{\delta}_{ij} = \frac{N_{ij}+N_{ji}}{2N}$
   $\hat{\theta}_{ij} = \frac{\hat{\delta}_{ij}}{\hat{\eta}_i \hat{\eta}_j} - 1$;

   b) Otherwise:
   $\hat{\delta}_{ij} = \frac{\sum(N_{pq}+N_{qp})}{2NK}$, where $\{p, q\}$ are the combinations of haplotypes that all result in the same genotype, and K is the number of such pairs;
   $\hat{\theta}_{ij} = \frac{\hat{\delta}_{ij}}{\hat{\eta}_i \hat{\eta}_j} - 1$.

5. Repeat 2-4 until the parameters no longer change.

Results

The root mean square errors over all iterations are shown in Table 3.2. Interesting to see is that the null model still performs well, just slightly worse than on data from a population that is in Hardy-Weinberg equilibrium.

The full model out-performs the null model, which indicates that the inclusion of $\hat{\theta}$ as a parameter allows for better estimations of $\hat{\eta}$. This was to be expected, but surprising is that the errors for $\hat{\theta}$ are quite high, considering most θ have a value between -1 and 1. This means that a (relatively) bad estimation of θ leads to a better estimation of η.

An explanation for this may be that the true value of θ in the sample is not representative of the true value of θ in the population, although the values for η are close to each other. In that case, the full model will accurately estimate the sample values of η and θ, but since the errors are measured against the population values, it ends up scoring better for η than θ.

Table 3.2: Simulation results scenario 2

		Null model		Full model		
# loci	N	rMSE($\hat{\eta}$)	rMSE($\hat{\delta}$)	rMSE($\hat{\eta}$)	rMSE($\hat{\theta}$)	rMSE($\hat{\delta}$)
2	100	.0369 (.0012)	.0170 (.0035)	.0304 (.0008)	.3009 (.0989)	.0178 (.0052)
2	500	.0165 (.0002)	.0132 (.0010)	.0134 (.0002)	.1230 (.0106)	.0081 (.0023)
3	100	.0266 (.0005)	.0051 (.0012)	.0183 (.0002)	.4278 (.0623)	.0075 (.0011)
3	500	.0133 (.0001)	.0033 (.0004)	.0114 (.0001)	.2290 (.0174)	.0030 (.0005)

3.3 Scenario 3: stratified population, known α

When a population is stratified and the mixing rate α is known, α can be used to accurately estimate $\hat{\eta}$ and $\hat{\theta}$ (see section 2.6). It is assumed that the subpopulations are in Hardy-Weinberg Equilibrium. Simulations are limited to genotype data on two loci, for populations with two strata, although the method can be generalised to more loci and strata (see section 2.6). Two values for α are used in the simulation: $\alpha = 0.5$ and $\alpha = 0.8$.

It is expected that the full model will out-perform the null model, since θ_{ij} are not equal to zero and α is specified correctly. Moreover, the full model can estimate the haplotype frequencies in the subpopulations, even when it is unknown which individuals belong to which subpopulation.

Simulation procedure

1. Simulate diplotype frequencies for both subpopulations:

$$\begin{aligned} \eta_i^s &\sim \text{Dir}() \qquad \text{for } i \in \{0, ..., (2^{Nloci} - 1)\}, s \in \{1, 2\} \\ \delta_{ij}^s &= \eta_i^s \eta_j^s \\ \left(\theta_{ij}^s \right.&\left.= 0\right) \end{aligned}$$

2. Calculate η and θ for the total population:

$$\begin{aligned} \eta_i &= \alpha\eta_i^1 + (1-\alpha)\eta_i^2 \\ \theta_{ij} &= \frac{\alpha\eta_i^1\eta_j^1 + (1-\alpha)\eta_i^2\eta_j^2}{(\alpha\eta_i^1 + (1-\alpha)\eta_i^2)(\alpha\eta_j^1 + (1-\alpha)\eta_j^2)} - 1 \end{aligned}$$

3. Simulate diplotype data for αN individuals using the diplotype frequencies of subpopulation 1 as probabilities and for $(1-\alpha)N$ individuals using the diplotype frequencies of subpopulation 2;

4. Combine the diplotype data into one population;

5. Convert diplotypes to genotype data.

Estimation procedure full model

1. Choose an initial value for $\hat{\theta}_{03}$;

2. Choose initial values for $\hat{\eta}$ and $\hat{\theta}$;

3. Calculate the expected likelihood for the data given the current parameter values;

4. Set new values for $\hat{\eta}$ based on the expected likelihood for the data;
5. Set new values for $\hat{\theta}$ using the values for $\hat{\eta}$ and the data;
6. Estimate $\hat{\eta}^1$ and $\hat{\eta}^2$ from $\hat{\eta}$ and α using numerical optimisation;
7. Recalculate $\hat{\theta}_{03}$ using $\hat{\eta}_0^1, \hat{\eta}_0^2, \hat{\eta}_3^1, \hat{\eta}_3^2$ and α;
8. Repeat 2-7 until the difference between the old and new estimate for $\hat{\theta}_{03}$ is smaller than some value ϵ.

Results

The simulation results are shown in Tables 3.3 and 3.4. Contrary to what was expected, the full model does not perform better than the new model - in fact, its errors are slightly larger. As in the previous scenario, the rMSE for $\hat{\theta}$ are large. This would again be explained if the parameters in the sample are not representative of those in the population. This theory is further explored in section 3.5. Another possible explanation is that the full model only performs better than the null model when the haplotype frequencies in the subpopulations are distinct enough. This possibility is written out in section 3.6.

Table 3.3: Simulation results scenario 3 - parameters total population

		Null model		Full model		
α	N	rMSE($\hat{\eta}$)	rMSE($\hat{\delta}$)	rMSE($\hat{\eta}$)	rMSE($\hat{\theta}$)	rMSE($\hat{\delta}$)
0.5	100	.0342 (.0009)	.0118 (.0047)	.0369 (.0009)	.3899 (.2816)	.0216 (.0063)
0.5	500	.0160 (.0002)	.0061 (.0024)	.0194 (.0004)	.1346 (.0171)	.0096 (.0033)
0.8	100	.0342 (.0010)	.0116 (.0050)	.0367 (.0010)	.2714 (.0560)	.0195 (.0054)
0.8	500	.0157 (.0002)	.0056 (.0023)	.0182 (.0004)	.1358 (.0214)	.0093 (.0032)

Table 3.4: Simulation results scenario 3 - parameters subpopulations

		Full model	
α	N	rMSE($\hat{\eta}^1$)	rMSE($\hat{\eta}^2$)
0.5	100	.1168 (.0101)	.1262 (.0109)
0.5	500	.0919 (.0089)	.0952 (.0087)
0.8	100	.0653 (.0034)	.2316 (.0399)
0.8	500	.0467 (.0020)	.1730 (.0234)

Interesting to note is that when $\alpha = 0.5$, the errors for the haplotype frequencies in both subpopulations are very similar, but when $\alpha = 0.8$, the errors are much smaller in the subpopulation with more individuals than in the other subpopulation. This is not very surprising, considering the case with $N = 100$ and $\alpha = 0.8$ only contains data on 20 individuals from subpopulation 2, therefore any estimate of the haplotype frequencies will not be very representative of the true subpopulation parameters.

3.4 Scenario 4: stratified population, misspecified α

This scenario explores how well the full model estimates the parameters when α is estimated incorrectly. The simulation and estimation procedures are the same as those laid out in section

3.3, with the exception that the α used in the estimation procedure is now the (incorrect) $\hat{\alpha}$. This scenario is simulated twice, once with ($\alpha = 0.5, \hat{\alpha} = 0.2$) and once with ($\alpha = 0.8, \hat{\alpha} = 0.2$).

Results

The simulation results are shown in Table 3.5. Since in the previous scenario, where α was correctly specified, the full model performed worse than the null model, it is unsurprising that the same result occurs here. What is interesting, however, is that an incorrectly chosen α does not seem to result in higher errors in the total population than a correct one. Rather, the difference in results compared to the previous scenario lies only in the errors for the estimates of the haplotype frequencies in the subpopulation.

The errors for subpopulation 1, which in reality contains 50% of the individuals but is thought to contain only 20%, are more than twice as large as the errors for subpopulation 2. This shows that the algorithm depends mostly on the number of individuals a subpopulation is estimated to contain, rather than how many individuals it actually contains. This makes sense, because none of the actual haplotype frequencies were set to zero, so any individual has a good chance of belonging to either subpopulation.

Table 3.5: Simulation results scenario 4 - parameters total population

			Null model		Full model		
α	α	N	rMSE($\hat{\eta}$)	rMSE($\hat{\delta}$)	rMSE($\hat{\eta}$)	rMSE($\hat{\theta}$)	rMSE($\hat{\delta}$)
0.5	0.2	100	.0342 (.0010)	.0117 (.0050)	.0369 (.0012)	.2617 (.0484)	.0195 (.0056)
0.5	0.2	500	.0162 (.0002)	.0061 (.0024)	.0196 (.0005)	.1376 (.0216)	.0097 (.0038)
0.8	0.2	100	.0349 (.0010)	.0119 (.0049)	.0378 (.0012)	.2789 (.0514)	.0201 (.0057)
0.8	0.2	500	.0156 (.0002)	.0055 (.0022)	.0192 (.0004)	.1397 (.0228)	.0097 (.0036)

Table 3.6: Simulation results scenario 4 - parameters subpopulations

		Full model	
α	N	rMSE($\hat{\eta}^1$)	rMSE($\hat{\eta}^2$)
0.5	100	.1993 (.0272)	.0841 (.0060)
0.5	500	.1551 (.0190)	.0647 (.0039)
0.8	100	.2009 (.0251)	.1078 (.0097)
0.8	500	.1430 (.0141)	.0912 (.0074)

3.5 Sample vs. population parameters

To explore the theory that some parameters in the sample may not be accurate estimates of the true population parameters, another simulation was performed. Diplotype frequencies are simulated for a stratified population with $\alpha = 0.5$, the same way as described in scenario 3. From the true diplotype frequencies δ^p, the true haplotype frequencies η^p and deviation parameters θ^p are calculated.

Next, diplotype data for these frequencies are simulated for sample sizes ranging from 100 to 10000. Since these diplotypes are not converted to genotypes, the sample diplotype frequencies in the sample δ^s are known, and the haplotype frequencies η^s and deviation parameters θ^s can be calculated directly. At each sample size, the rMSE for the three parameter sets are calculated:

$$rMSE(\delta, N) = \frac{(\delta^p - \delta^s)^2}{N}$$
$$rMSE(\eta, N) = \frac{(\eta^p - \eta^s)^2}{N}$$
$$rMSE(\theta, N) = \frac{(\theta^p - \theta^s)^2}{N}$$

This procedure is repeated 100 times. The average rMSE for each sample size is shown in Figure 3.1. Note that the rMSE on the vertical axis is shown on a log-scale. It can be seen that while the diplotype frequencies in the sample are very close to those in the population even at small sample sizes, the same does not hold for the deviation parameters. This can be explained by looking back at the definition of θ:

$$\theta_{ij} = \frac{\delta_{ij}}{\eta_i \eta_j} - 1$$

The problem lies in the fact that θ is calculated from η, which is calculated from δ, which is the only parameter that is estimated directly from the data. The errors in the parameters therefore add up when biased parameters are used to calculate other parameters, leading to very high errors for θ.

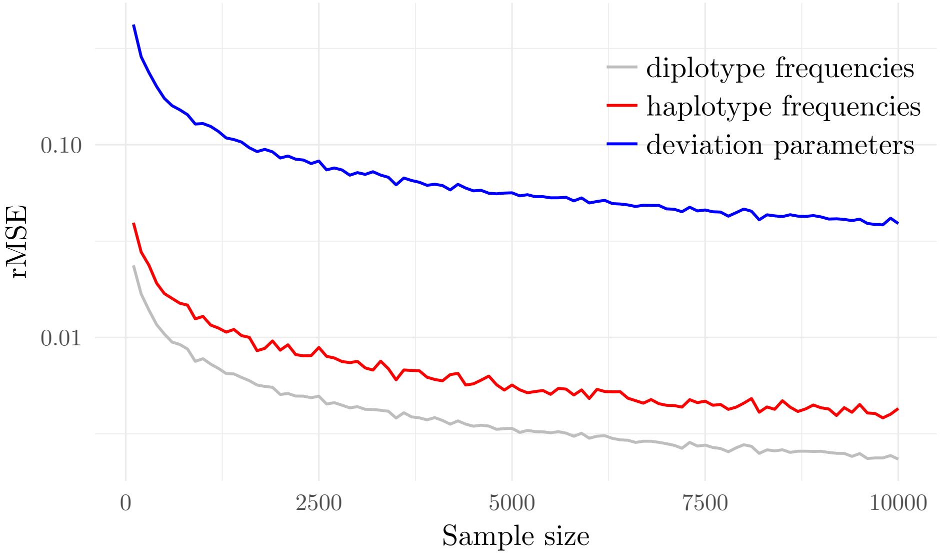

Figure 3.1: Root Mean Square Error of the parameters in the sample compared to the parameters in the population. The error is shown on the log-scale.

At $N = 100$, rMSE(θ) is approximately 0.43, and at $N = 500$ it is still around 0.2. This supports the theory that at these sample sizes, the estimation procedure for stratified populations will not be accurate, since the parameters it estimates are not representative for the populations. rMSE(θ) dips below 0.1 around $N = 1000$. When simulation scenario 3 is repeated for larger sample sizes, the results shown in Table 3.7 are obtained. Only 50 instead of 500 repetitions were performed to offset the longer computation time needed for larger sample sizes.

Table 3.7: Simulation results scenario 3 - larger N

		Null model		Full model		
α	N	rMSE($\hat{\eta}$)	rMSE($\hat{\delta}$)	rMSE($\hat{\eta}$)	rMSE($\hat{\theta}$)	rMSE($\hat{\delta}$)
0.5	1000	.0118 (.0001)	.0049 (.0020)	.0172 (.0004)	.1109 (.0169)	.0079 (.0037)
0.5	5000	.0056 (.0000)	.0030 (.0014)	.0072 (.0001)	.0471 (.0025)	.0033 (.0014)
0.8	1000	.0098 (.0001)	.0039 (.0015)	.0128 (.0002)	.1019 (.0111)	.0067 (.0025)
0.8	5000	.0053 (.0000)	.0025 (.0013)	.0066 (.0000)	.0484 (.0024)	.0034 (.0011)

While increasing the sample size results in much lower rMSE for the full model, the same holds for the null model, which in fact starts out-performing the full model even more. However, the full model can now estimate the haplotype frequencies in the subpopulations very accurately, which the null model cannot do. It is clear that the low sample size in simulation scenario 3 is not the (only) reason that the full model performs worse than expected.

3.6 Scenario 5: stratified population, contrasting haplotype frequencies

In scenarios 3 and 4, haplotype frequencies in the subpopulations were randomly simulated from a Dirichlet distribution. In this scenario, the models will be applied to subpopulations with very different (in fact, opposite) haplotype frequencies. The population parameters are not randomly simulated, but kept constant:

$$\eta^1 = \{0.1, 0.4, 0.4, 0.1\}$$
$$\eta^2 = \{0.4, 0.1, 0.1, 0.4\}$$
$$\eta = \{0.25, 0.25, 0.25, 0.25\}$$

Data was generated for these parameters for sample sizes 100, 500, 1000 and 5000, and the null and full model were used to estimate the parameters. The procedure was repeated 50 times for each sample size, and the average rMSE and standard deviation per parameter is shown in Table 3.8.

Table 3.8: Simulation results scenario 5

	Null model		Full model		
N	rMSE($\hat{\eta}$)	rMSE($\hat{\delta}$)	rMSE($\hat{\eta}$)	rMSE($\hat{\theta}$)	rMSE($\hat{\delta}$)
100	.0419 (.0234)	.0268 (.0041)	.0505 (.0258)	.3568 (.1568)	.0272 (.0100)
500	.0194 (.0092)	.0234 (.0008)	.0243 (.0132)	.1611 (.1244)	.0129 (.0071)
1000	.0134 (.0072)	.0230 (.0005)	.0258 (.0207)	.1840 (.1798)	.0134 (.0101)
5000	.0066 (.0038)	.0226 (.0001)	.0170 (.0205)	.1219 (.1866)	.0086 (.0109)

Interesting to see is that with increasing sample size, the null model keeps improving its estimates for the haplotype frequencies, but the estimates for the diplotype frequencies are almost constant. The full model, however, keeps improving both estimates with increasing sample size. Although the estimates for the haplotype frequencies are always more accurate in the null model, the full model out-performs the null model in terms of accuracy for the diplotype frequency. The difference between the models gets more pronounced at larger sample sizes, as the full model keeps improving while the null model does not.

3.7 Conclusions

The models proposed in this thesis are able to accurately estimate haplotype frequencies in different scenarios, although their added value over existing estimation models varies by scenario.

When Hardy-Weinberg equilibrium holds, simulation scenario 1 shows that the new models do not present an advantage over existing models, and in fact should rather not be used because of increased computation time.

For populations not in Hardy-Weinberg equilibrium, two methods have been explored and tested. When no information about the composition of a stratified population is available, or the population is thought to be outside of the equilibrium for reasons other than stratification, heuristics on the diplotype frequencies can be used. The full model in simulation scenario 2 assumes all diplotypes that lead to the same genotype to have the same frequency. This assumption results in lower errors for the haplotype frequencies, although the difference is not very large. Better results may be expected when the estimation for the diplotype frequencies are based on previous research, as in this simulation it was purely random.

The remaining scenarios explored the accuracy of the proposed model that can be used when the researcher is able to make an assumption about the mixing rate of a stratified population (consisting of two strata in which HWE holds). When haplotype frequencies for the subpopulations are randomly simulated, the full model results in good estimates for the haplotype frequencies in the total population, although it performs slightly worse than the null model. The errors for the haplotype frequencies in the subpopulations are significantly higher than those for the total population, especially at low sample sizes. If the specified mixing rate is incorrect, the haplotype frequencies in the total population are still estimated accurately, but the estimates for the haplotype frequencies in the subpopulation have much higher errors.

Simulation scenario 5 showed that the real added value of the proposed algorithm lies in stratified populations of which the subpopulations have very different haplotype frequencies. When this is the case, the estimates for the diplotype frequencies are better than those obtained from models that assume HWE, and the difference increases at higher sample sizes.

The conclusion from testing the proposed algorithm on simulated data sets is that it never performs much worse than existing algorithms, but when subpopulations have haplotype frequencies that are not close together, it performs much better. Therefore, when this is thought to be the case, using the proposed algorithm is advisable, as the only real cost is that the model takes more computation time.

Chapter 4

HapMap data analysis

This chapter applies the method to estimate haplotype frequencies in stratified populations on the HapMap phase II data set. The International HapMap Project aimed to develop a haplotype map of the human genome as a public resource to accelerate medical genetic research. The project discovered millions of new SNPs, and the data was used in many genome-wide association studies to search for disease-associated SNPs.

4.1 Data set

The phase II data set was published in October 2007 [3]. It contains genotype data of 270 individuals on over 3.1 million SNPs. The individuals come from four geographically diverse populations:

- YRI (N = 90): 30 mother-father-child trios from the Yoruba in Ibadan, Nigeria;
- CEU (N = 90): 30 trios of individuals of northern and western European ancestry, living in Utah;
- CHB (N = 45): 45 unrelated Han Chinese individuals in Beijing, China;
- JPT (N = 45): 45 unrelated Japanese individuals in Tokyo, Japan.

The HapMap phase II data contains approximately one SNP per one kilobase, an estimated 25-35% of all SNPs in the human genome with a minor allele frequency larger than 5%.

4.2 Region selection

Three different regions of SNPs were selected to apply the model to. The UCSC Genome Browser [4] was used to search for positions of SNPs on chromosomes. These SNPs were then selected from the entire HapMap phase II data set to perform the analysis on.

The regions were selected because of the very differing ways natural selection acts (or doesn't act) on them. The first region encodes a gene that has many different alleles, and which favours high genetic variability. The second region contains a gene that is regulated strictly and shows very low variation. The third region is a non-coding region where SNPs are not expected to be subject to selection at all.

Chromosome 6: HLA-A gene

The first region to be analysed is the HLA-A gene on the short arm of chromosome 6. HLA stands for human leukocyte antigen and is a major histocompatibility complex (MHC) class I antigen that only occurs on humans [5]. HLA-A is one of the three major MHC types, next to HLA-B and HLA-C, which are also encoded on the short arm of chromosome 6. Hundreds of HLA-A alleles have been described, and the genetic variation within populations is usually high. Each allele binds to different peptide structures, and thus provides resistance against different pathogens. The high variation within the population therefore protects the population from mass extinction due to a single invader.

Studies indicate that in humans as well as certain animals the genetic variation in HLA genes is maintained by a heterozygous selection mechanism. Some of the hypothesised mechanisms are mate choice based on pheromones or allele-dependent spontaneous abortion [6] [7].

Chromosome 7: HOX-A10 gene

The second region is a gene on the short arm of chromosome 7 that codes for homeobox protein HOX-A10. Homeoboxes are DNA sequences within genes that play an important role in anatomical development during the embryonic stage. Mutations in these genes often lead to phenotypic changes, and duplication can even increase new body segments in insects. In humans, these genes are highly conserved and mutations are rare.

HOX-A10 specifically may have a function in fertility and embryo viability [5]. Studies have also shown that it plays a role in blood cell differentiation [8].

Chromosome 8: gene desert

The last selected region is the gene desert on the long arm of chromosome 8: the 8q24 gene region. Gene deserts are stretches of DNA that are entirely devoid of protein-coding genes, and they are estimated to make up 25% of the human genome. Even though these regions of DNA are non-coding, many SNPs within them have been found to be associated with genetic diseases. The 8q24 gene desert is linked to increased risks for prostate, breast, ovarian, colonic and pancreatic cancer [9].

4.3 Method

Each region described in the previous section was analysed separately. The SNP locations corresponding to the regions were collected using the online Genome Browser [4]. The locations were then used to extract the appropriate SNPs from the entire HapMap phase II data set, which also includes the population for each individual. All analyses were performed in R.

In each region, a sliding window approach was applied to get haplotype frequencies. First, haplotype frequencies for pairs of SNPs were calculated within each population separately. This was done using the existing algorithm that assumes the (sub)population is in Hardy-Weinberg equilibrium.

Next, the aim was to compare the populations from Utah (CEU) and Nigeria (YRI) to each other, as these populations have the highest sample size and can be assumed to be genetically diverse, since they are geographically separated. All pairs of SNPs for which every possible haplotype occurred at least once in either the CEU or the YRI population were submitted to the algorithm described in section 2.8.

For each of those SNP pairs, the Euclidean distance between the haplotype frequencies of both subpopulation was calculated. This was done to see how diverse the subpopulations are, and if that influences the accuracy of the algorithm. The Euclidean distance is calculated as follows:

$$d(\eta^1, \eta^2) = \sqrt{\sum_{i=0}^{3} (\eta_i^2 - \eta_i^1)^2} \tag{4.1}$$

4.4 Results

Chromosome 6: HLA-A gene

In total, genotype data on 25 SNPs was available in the data set for this region. The haplotype frequencies for each SNP pair per population can be found in Appendix C. Of those 24 SNP pairs, there were 16 for which each haplotype frequency was non-zero in either the CEU or the YRI population.

For each pair of SNPs, two sets of rMSE were calculated: one assuming that the first subpopulation returned by the algorithm was the CEU population and the second the YRI population, and one vice versa. The set with the lowest total rMSE was assumed to be the correct one, and those values are shown in Table 4.1. Also shown are the rMSE for the total population (the total population here being the combination of the CEU and YRI subpopulations), and the Euclidean distance between the true haplotype frequencies in the subpopulations.

Table 4.1: Results chromosome 6: HLA-A gene

SNP 1	SNP 2	rMSE($\hat{\eta}^1$)	rMSE($\hat{\eta}^2$)	rMSE($\hat{\eta}$)	d(η^1, η^2)
rs2523793	rs2735020	.2229	.0619	.0244	.5497
rs2735020	rs2517889	.1620	.0998	.0273	.5102
rs2523790	rs16896049	.1394	.1346	.0113	.5468
rs1233326	rs9404952	.0337	.0836	.0345	.1822
rs9404952	rs16896052	.0645	.0024	.0026	.1336
rs16896052	rs2523786	.1522	.1280	.0168	.5591
rs2394179	rs2394180	.2670	.1610	.1007	.5522
rs2523783	rs2394182	.0883	.1672	.0602	.4787
rs2735015	rs2735014	.0361	.3320	.1279	.6143
rs2735014	rs9468641	.1348	.1440	.0024	.5568
rs9468641	rs2253981	.0680	.1828	.0027	.5014
rs2253981	rs7755504	.1772	.0626	.0410	.4602
rs7755504	rs2254071	.0287	.2134	.0419	.4602
rs2254077	rs2523781	.1562	.1216	.0588	.4935
rs2523781	rs2523780	.2822	.0340	.0588	.5439
rs2523780	rs2523779	.2500	.0553	.0492	.5537
	average	.1414 (.0839)	.1240 (.0780)	.0413 (.0352)	

Chromosome 7: HOX-A10 gene

In total, genotype data on 50 SNPs was available in the data set for this region. The haplotype frequencies for each SNP pair per population can be found in Appendix C. Of those 49 SNP pairs, there was only one pair for which each haplotype frequency was non-zero in either the CEU or the YRI population. Moreover, many SNPs did not show any variation at all, as all individuals expressed the exact same genotype. Table 4.2 shows the rMSE for both subpopulations and the Euclidean distance between the true haplotype frequencies in the subpopulations.

Table 4.2: Results chromosome 7: HOX-A10 gene

SNP 1	SNP 2	rMSE($\hat{\eta}^1$)	rMSE($\hat{\eta}^2$)	rMSE($\hat{\eta}$)	d(η^1, η^2)
rs3735533	rs10228276	.0981	.1114	.0207	.4174

Chromosome 8: gene desert

In total, genotype data on 153 SNPs was available in the data set for this region. The haplotype frequencies for each SNP pair per population can be found in Appendix C. Of those 152 SNP pairs, there were 18 for which each haplotype frequency was non-zero in either the CEU or the YRI population. Table 4.3 shows the rMSE for all SNP pairs in both subpopulations and the total population and the Euclidean distance between the true haplotype frequencies in the subpopulations.

Table 4.3: Results chromosome 8: gene desert

SNP 1	SNP 2	rMSE($\hat{\eta}^1$)	rMSE($\hat{\eta}^2$)	rMSE($\hat{\eta}$)	d(η^1, η^2)
rs11786769	rs17607388	.0301	.0229	.0072	.0982
rs4909622	rs4243860	.2567	.1390	.1432	.2850
rs10454360	rs10112787	.1384	.0274	.0013	.2871
rs4481635	rs10088784	.1075	.2651	.1269	.3218
rs10088784	rs10505668	.2617	.1559	.1246	.3211
rs11994631	rs4517157	.0163	.0393	.0129	.1108
rs4517157	rs4645591	.0945	.0956	.0082	.3770
rs11784156	rs7821320	.2468	.1150	.0820	.6852
rs4295697	rs3903129	.1068	.2006	.1162	.3715
rs3903129	rs7819274	.2197	.0596	.0874	.4170
rs6991281	rs2076987	.1863	.0723	.0352	.4716
rs4256627	rs7831122	.2702	.1959	.1550	.6389
rs7831122	rs6577754	.2326	.0996	.1565	.6631
rs16906764	rs1014197	.0912	.0989	.0555	.2964
rs16906771	rs16906772	.0468	.0942	.0670	.2049
rs11987180	rs11990934	.1014	.0355	.0245	.1636
rs4319134	rs4527908	.1314	.2155	.0013	.6685
rs7827430	rs7831697	.1289	.2339	.1390	.7088
	average	.1482 (.0830)	.1203 (.0755)	.0747 (.0580)	

4.5 Conclusions

The average rMSE for the subpopulations and the 'total population' (YRI and CEU combined) are consistent among the three regions, and are similar to the errors found in simulated data sets. The algorithm performs well at estimating the total haplotype frequencies, it misses the true value by about 5 percentage points on average. Considering the total sample size is only 180, these are reasonable estimates.

The errors for the haplotype frequencies in the subpopulation are significantly higher: they miss the true value by 12 to 14 percentage points on average. This can be explained by two factors. First, the algorithm does not know which individual belongs to which subpopulation, and most individuals can not be placed in a subpopulation with 100% certainty, unless they express a haplotype that occurs in only one subpopulation. Second, the sample size decreases by 50% when going from estimating parameters in the total population to the subpopulation, which also leads to higher errors.

There are significant differences between SNP pairs with regards to errors in the subpopulations. The SNP pair that gets the best estimation is (rs11786769, rs17607388), the first pair in the third region (chromosome 8, gene desert). The haplotype frequencies in the subpopulations are on average within three percentage points of the true value, which is very accurate at these sample sizes. Looking at the haplotype frequencies in the subpopulations (see Tables C.9 and C.10), an explanation for this good score may be that the homozygous recessive haplotype does not occur in the CEU population, but does in the YRI population. The same holds for other SNP pairs that have very low errors in the subpopulations: one or more haplotypes occur in only one subpopulation. This makes the subpopulations easier to identify for the algorithm, resulting in lower errors.

Figure 4.1 shows for each SNP pair the Euclidean distance between the haplotype frequencies

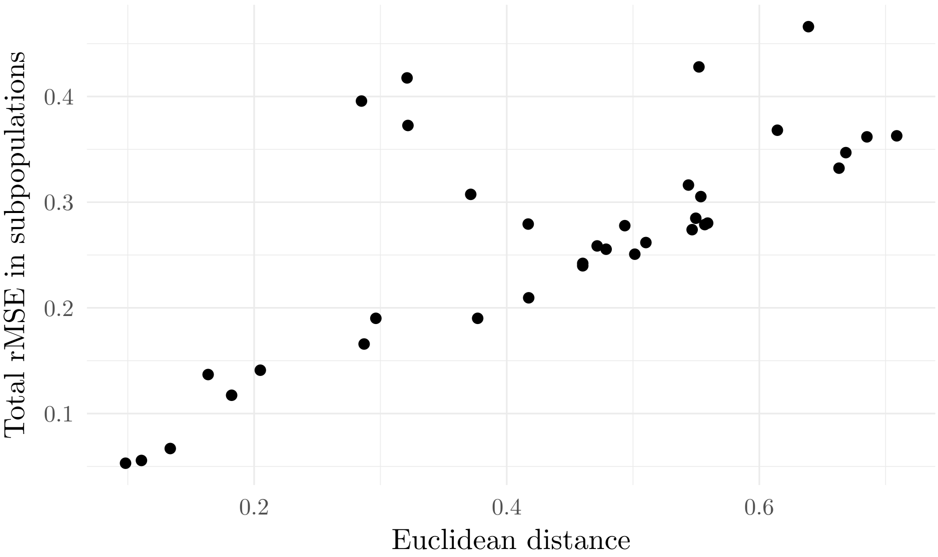

Figure 4.1: Root Mean Square Error of haplotype frequencies in the subpopulations against the Euclidean distance between the subpopulations. The vertical axis shows the sum of rMSE($\hat{\eta}^1$) and rMSE($\hat{\eta}^2$).

in the subpopulations, and the total rMSE for the haplotype frequencies. A larger distance between the subpopulations leads to higher errors. This may be because incorrectly placing an individual in a subpopulation has a larger influence when the haplotype frequencies in the subpopulations are further apart, leading to higher errors. The special case mentioned in the previous paragraph, where one or more haplotypes occur in only one of the subpopulations, would be an exception to this, as this will increase the Euclidean distance but decrease the total rMSE.

As mentioned before, the regions were selected because of the different ways natural selection acts (or does not act) on these stretches of DNA. These differences can also be seen in the results of the data analysis. The HLA-A gene is thought to be subject to a selection mechanism that promotes genetic diversity, and for 16 out of 24 SNP pairs all haplotypes are present in the total population (CEU and YRI combined, in this case). This is a stark contrast with the HOX-A10 gene, which is very conservative, and for which this was only the case for 1 out of 49 SNP pairs. The last region, the 8q24 gene desert, contained 152 SNP pairs in total, and 18 of those had non-zero frequencies for all haplotypes. This is relatively a lot less than in the HLA-A region, where variability is actively maintained, and less than would have been expected if there were no selection mechanisms at play at all in the region. This result is in line with results from many studies that found SNP associations with genetic diseases in this region. For those SNPs, certain haplotypes might be selected for or against, while other SNPs that are not associated with any phenotypic trait may not be subject to selection at all.

Chapter 5

Conclusion and discussion

In this thesis, a new algorithm was described to estimate haplotype and diplotype frequencies in stratified populations. Existing algorithms assume that the populations they are applied to are in Hardy-Weinberg equilibrium, but stratified populations are not. When deviations from Hardy-Weinberg equilibrium are added to the existing models, they become unidentifiable as the number of parameters to be estimated from the data is too large. Constraints need to be added to the model in order to uniquely estimate the parameters.

The proposed algorithm introduces yet another parameter to the parameter space. However, this parameter is assumed to be known, and with it, two existing parameters can be calculated. This leads to a decrease in the number of parameters to be estimated, which makes the model identifiable again.

Simulations

The new algorithm was applied to a number of simulated data sets and compared to the existing algorithm. These simulation scenarios showed when the new algorithm provides an advantage over the existing one. The conclusion of these simulations was that the new algorithm outperforms the old one in terms of accuracy of estimated diplotype frequencies only when it is applied to stratified populations for which the subpopulations have very different haplotype frequencies. The reason for this is that the model does not know which individual belongs to which subpopulation, and when the haplotype frequencies are close together, the likelihoods of belonging to either subpopulation are nearly the same. However, when haplotype frequencies are far apart, it becomes easier to estimate to which subpopulation an individual belongs, and therefore the estimations of the haplotype frequencies in the subpopulations become more accurate.

HapMap data

Next, the algorithm was applied to a real data set, from the International HapMap project. This data set contains genotype information of individuals from four geographically diverse populations. Three regions in the genome were explored: the HLA-A gene on chromosome 6, in which a heterozygous selection mechanism is thought to operate; the HOX-A10 gene on chromosome 7, which is a very conservative region of DNA; and the 8q24 gene desert on chromosome 8, which is a stretch of non-coding DNA that contains many SNPs that have been found to be associated with genetic diseases.

The analysis of these regions illustrated the differences between them, and also when the new algorithm is most useful. It works best on stretches of DNA that show high variability between populations, especially when some haplotypes occur in only one of the subpopulations.

Difficulties

Several difficulties were encountered during this thesis. The basis of the proposed algorithm lies in introducing a deviation parameter for each diplotype, and using these parameters to estimate haplotype and diplotype frequencies more accurately. However, it turns out that the deviation parameters in the sample exhibit very large sampling variation. Although asymptotically they will converge with the population parameters, this requires large sample sizes ($N >> 5000$), which is almost never feasible with real data sets. This causes the estimations of the new algorithm to have higher errors than those of existing algorithms, even though in theory they should fit the data better. Only when subpopulations are very distinct, the new algorithm provides an advantage over existing ones. It might be worth exploring whether the deviation parameters can be defined in a different way, so that they are not dependent on biased estimates of other parameters, but rather estimated directly from the data. This will reduce the errors in the parameters, and increase the usefulness of the new model in a wider range of scenarios.

Another problem that was encountered is that during the estimation process in the new algorithm, a local maximum might be found instead of a global one, which also leads to worse estimations. This occurs during the estimation of the haplotype frequencies in the subpopulation, which is done using numerical optimisation. It also sometimes occurred that the optimisation function got 'stuck' in a positive feedback loop because of a bad starting value, leading to estimated negative diplotype frequencies, which are of course impossible. This was sidestepped by allowing a range of different starting values and ignoring outcomes that resulted in impossible frequencies. However, both these problems may be solved by replacing the numerical optimisation by an algebraic method.

It would be useful to get not only estimates for the haplotype frequencies, but also estimates for their standard errors. Assuming the true populations are unknown, this might be done using a form of cross-validation, by splitting the population into several folds and calculating the parameters in each fold.

This method can also be generalised to more loci and more subpopulations. The easiest of those is more loci: the generalisation has been discussed in section 2.2. The optimisation process works exactly the same for more loci. To apply the method to more than two subpopulations, an iterative process should be used, with a leave-one-out mechanism. Group all but one subpopulations together, and run the algorithm for one subpopulation versus the rest to get the haplotype frequencies. Repeat this for each subpopulation to get estimates for all the parameters.

Possible applications

The proposed algorithm will be useful for any research area that deals with genetic data and distinct subpopulations, and want to study haplotypes. Genetic disease studies may favour this algorithm over existing ones when dealing with diseases that are more common in certain continents or populations. Estimating the haplotype frequencies in the subpopulations more accurately will also allow for better predictions when an individual's genotype is converted to a diplotype. In disease studies, the subpopulations may be the case and control group, which may or may not express very different haplotype frequencies.

Population geneticists may also find the algorithm useful to infer haplotypes from genotype data without the added cost of sequencing the haplotypes. Especially on large scales, computational methods are a cheap alternative for haplotype sequencing, but if the method is not very accurate the financial savings may not be worth sacrificing accuracy. Allowing populations to be outside of Hardy-Weinberg equilibrium means that a wider range of populations can be subjected to an algorithm to infer haplotypes from genotypes.

The algorithm can also be used as an exploratory tool. If there is no information on stratification in the data set, this algorithm can be run to explore whether or not it finds distinct subpopulations. The haplotype frequencies in the subpopulations that are returned by the algorithm can be compared to see if they are actually distinct. An extension of this algorithm might be to add a predictive function that assigns each individual a likelihood of belonging to each subpopulation. Then, these subpopulations might be investigated to see whether they share certain non-genotypic traits, such as gender or descent. The presence of such shared traits would be further evidence that the subpopulations found by the algorithm are valid.

Bibliography

[1] Jane B Reece et al. *Campbell biology.* Vol. 9. Pearson Boston, 2011.

[2] Arthur P Dempster, Nan M Laird, and Donald B Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[3] International HapMap Consortium et al. “A second generation human haplotype map of over 3.1 million SNPs”. In: *Nature* 449.7164 (2007), p. 851.

[4] W James Kent et al. “The human genome browser at UCSC”. In: *Genome research* 12.6 (2002), pp. 996–1006.

[5] Nuala A O’Leary et al. “Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation”. In: *Nucleic acids research* 44.D1 (2015), pp. D733–D745.

[6] Victor Apanius et al. “The nature of selection on the major histocompatibility complex”. In: *Critical Reviews in Immunology* 17.2 (1997).

[7] Claus Wedekind et al. “MHC-dependent mate preferences in humans”. In: *Proc. R. Soc. Lond. B* 260.1359 (1995), pp. 245–249.

[8] H Jeffrey Lawrence et al. “The role of HOX homeobox genes in normal and leukemic hematopoiesis”. In: *Stem Cells* 14.3 (1996), pp. 281–291.

[9] Maya Ghoussaini et al. “Multiple loci with different cancer specificities within the 8q24 gene desert”. In: *JNCI: Journal of the National Cancer Institute* 100.13 (2008), pp. 962–966.

Appendix A

Conditional distribution of diplotype frequencies - 3 loci

Consider data of genotypes on three loci, with alleles $\{A, a\}$ on locus 1, $\{B, b\}$ on locus 2 and $\{C, c\}$ on locus 3. The possible haplotypes and their frequencies are:

$$\begin{aligned}
\eta_0 &= P(\{abc\}) \\
\eta_1 &= P(\{abC\}) \\
\eta_2 &= P(\{aBc\}) \\
\eta_3 &= P(\{aBC\}) \\
\eta_4 &= P(\{Abc\}) \\
\eta_5 &= P(\{AbC\}) \\
\eta_6 &= P(\{ABc\}) \\
\eta_7 &= P(\{ABC\})
\end{aligned}$$

All frequencies must sum to one: $\sum \eta_i = 1$. The population is not in Hardy-Weinberg equilibrium, so the diplotype frequencies are calculated by multiplying the haplotype frequencies by a deviation parameter. As in 2.2, symmetry is assumed.

$$\delta_{ij} = \delta_{ji} = \eta_i \eta_j (1 + \theta_{ij})$$

The genotype score for individual i is written as: $g_i = (g_{i1}, g_{i2}, g_{i3})$. Here, a genotype is coded as 0 when it is homozygous recessive, 1 when it is heterozygous and 2 when it is homozygous dominant. In other words, the genotype score can be interpreted as the number of dominant alleles at each locus.

Depending on the observed genotype, we can distinguish four cases for the diplotype:

1. All three genotypes are homozygous. In this case, the unordered diplotype is unique.

2. One genotype is heterozygous. As in the previous case, the unordered diplotype is unique.

3. Two genotypes are heterozygous. Now, two unordered diplotypes are possible. Example: genotype $\{AaBbCC\}$ can have either haplotype $\{ABC/abC\}$ or haplotype $\{AbC/aBC\}$. Any two of the three loci can be heterozygous, and the homozygous locus can be either recessive or dominant, so there are six ways this case can occur.

4. All three genotypes are heterozygous. In this case, four unordered diplotypes are possible. Example: genotype $\{AaBbCc\}$ can have the following unordered haplotypes:

   $\{abc/ABC\}, \{abC/ABc\}, \{aBc/AbC\}, \{Abc/aBC\}$.

This translates into the following conditional distribution for data $D = (d_1, ..., d_N)$ (unordered diplotypes):

$$P(d_i|g_i, \eta, \theta) = \left\{ \begin{array}{ll} 0, & d_i \notin \phi^{-1}(g_i), \\ 1, & \text{all } g_i \in \{0, 2\} \\ \hline \eta_0\eta_3(1+\theta_{03})/\pi_{g_i}, & g_i = (0,1,1), d_i \in \{(0,3),(3,0)\} \\ \eta_1\eta_2(1+\theta_{12})/\pi_{g_i}, & g_i = (0,1,1), d_i \in \{(1,2),(2,1)\} \\ \eta_0\eta_5(1+\theta_{05})/\pi_{g_i}, & g_i = (1,0,1), d_i \in \{(0,5),(5,0)\} \\ \eta_1\eta_4(1+\theta_{14})/\pi_{g_i}, & g_i = (1,0,1), d_i \in \{(1,4),(4,1)\} \\ \eta_0\eta_6(1+\theta_{06})/\pi_{g_i}, & g_i = (1,1,0), d_i \in \{(0,6),(6,0)\} \\ \eta_2\eta_4(1+\theta_{24})/\pi_{g_i}, & g_i = (1,1,0), d_i \in \{(2,4),(4,2)\} \\ \hline \eta_1\eta_7(1+\theta_{17})/\pi_{g_i}, & g_i = (1,1,2), d_i \in \{(1,7),(7,1)\} \\ \eta_3\eta_5(1+\theta_{35})/\pi_{g_i}, & g_i = (1,1,2), d_i \in \{(3,5),(5,3)\} \\ \eta_2\eta_7(1+\theta_{27})/\pi_{g_i}, & g_i = (1,2,1), d_i \in \{(2,7),(7,2)\} \\ \eta_3\eta_6(1+\theta_{36})/\pi_{g_i}, & g_i = (1,2,1), d_i \in \{(3,6),(6,3)\} \\ \eta_4\eta_7(1+\theta_{47})/\pi_{g_i}, & g_i = (2,1,1), d_i \in \{(4,7),(7,4)\} \\ \eta_5\eta_6(1+\theta_{56})/\pi_{g_i}, & g_i = (2,1,1), d_i \in \{(5,6),(6,5)\} \\ \hline \eta_0\eta_7(1+\theta_{07})/\pi_{g_i}, & g_i = (1,1,1), d_i \in \{(0,7),(7,0)\} \\ \eta_1\eta_6(1+\theta_{16})/\pi_{g_i}, & g_i = (1,1,1), d_i \in \{(1,6),(6,1)\} \\ \eta_2\eta_5(1+\theta_{25})/\pi_{g_i}, & g_i = (1,1,1), d_i \in \{(2,5),(5,2)\} \\ \eta_3\eta_4(1+\theta_{34})/\pi_{g_i}, & g_i = (1,1,1), d_i \in \{(3,4),(4,3)\} \\ \hline 1, & else \end{array} \right.$$

The π_{g_i} are calculated as $\pi_{g_i} = \sum_h P(h, g_i|\eta, \theta)$. For example:

$$\begin{aligned} \pi_{g_i=(0,1,1)} &= 2(\eta_0\eta_3 + \eta_1\eta_2) \\ \pi_{g_i=(1,1,1)} &= 2(\eta_0\eta_7 + \eta_1\eta_6 + \eta_2\eta_5 + \eta_3\eta_4) \end{aligned}$$

The factor 2 is included to account for the fact that each unordered diplotype has two possible ordered haplotype, with parental origins swapped.

Appendix B

R code

B.1 Data simulation

```
library(gtools)

simDtfs <- function(Nloci = 2, HWE = FALSE, beta = 1e3) {
  M <- 2^Nloci
  if (HWE) {
    h <- rdirichlet(1, rep(1, M) * beta)
    dtfs <- t(h) %*% h
  } else {
    d <- rdirichlet(1, as.vector(diag(rep(0.4, M)) + 0.8) * beta)
    dtfs <- matrix(d, M, M)
    for (i in 1:M) {
      for (j in 1:M) {
        if (i < j) {
          dtfs[i, j] <- mean(c(dtfs[i, j], dtfs[j, i]))
          dtfs[j, i] <- dtfs[i, j]
        }
      }
    }
  }
  return(dtfs)
}

simDtfsStrat <- function(Nloci = 2, Nstrat = 2, alpha, beta) {
  M <- 2^Nloci
  h1 <- rdirichlet(1, rep(1,M) * beta)
  dtfs1 <- t(h1) %*% h1
  h2 <- rdirichlet(1, rep(1,M) * beta)
  dtfs2 <- t(h2) %*% h2
  dtfs <- (alpha * dtfs1 + (1-alpha) * dtfs2)
  return(list(dtfs = dtfs, htfs1 = h1, htfs2 = h2))
}

calcHtfs <- function(dtfs) {
  htfs <- rowSums(dtfs)
  return(htfs)
}

calcThetas <- function(dtfs, htfs) {
  M <- length(htfs)
  thetas <- matrix(0, M, M)
  for (i in 1:M) {
    for (j in 1:M) {
      thetas[i, j] <- dtfs[i, j] / (htfs[i] * htfs[j]) - 1
    }
  }
  return(thetas)
}

gtsForDtfs <- function(dtfs, N) {
  Nloci <- log2(ncol(dtfs))
  M <- 2^Nloci
  dtfsv <- as.vector(dtfs)
  dtn <- character()
  for (i in 1:M^2) {
    dtn <- c(dtn, (paste0(as.matrix(expand.grid(1:M, 1:M))[i, 2],
                          as.matrix(expand.grid(1:M, 1:M))[i, 1])))
  }
  dtsc <- sample(dtn, N, replace = TRUE, prob = dtfsv)
  hts <- matrix(0, N, 2)
  for (i in 1:N) {
    hts[i, ] <- c(as.numeric(substr(dtsc[i], 1, 1)),
                  as.numeric(substr(dtsc[i], 2, 2)))
  }
  gts <- matrix(NA, N, Nloci)
  for (i in 1:N) {
    gts[i, ] <- ord2bin(hts[i, 1], digits = Nloci) +
                ord2bin(hts[i, 2], digits = Nloci)
  }
  return(gts)
}

simGts <- function(N, Nloci = 2, HWE = FALSE, strat = F, Nstrat = 2,
                   alpha = alpha, beta = 1e3) {
  if (strat == TRUE) {
    out <- simDtfsStrat(Nloci = Nloci, Nstrat = Nstrat,
                        alpha = alpha, beta = beta)
    dtfs <- out$dtfs
    htfs1 <- out$htfs1
    htfs2 <- out$htfs2
  } else {
    dtfs <- simDtfs(Nloci = Nloci, HWE = HWE, beta = beta)
    htfs1 <- NULL
    htfs2 <- NULL
  }
  htfs <- calcHtfs(dtfs)
  thetas <- calcThetas(dtfs, htfs)
  gts <- gtsForDtfs(dtfs, N)
  return(list(dtfs = dtfs, htfs = htfs, thetas = thetas, gts = gts,
              htfs1 = htfs1, htfs2 = htfs2))
}
```

B.2 EM algorithm

```
ord2bin = function(o, digits = 2) {
  b <- rev(as.numeric(intToBits(o-1)))
  b <- b[-(1:(length(b) - digits))]
  return(b)
}

bin2ord <- function(b, digits = 2) {
  o <- 1
  for (i in 1:digits) {
    o <- o + rev(b)[i] * 2 ^ (i-1)
  }
  return(o)
}

prob_dt_given_gt <- function(dt, gt, htfs, thetas = matrix(0, 16, 16)) {
  # Returns probabilities of diplotypes, given genotypes
  # in:  dt     : diplotype as vector of 2 integers
  #      gt     : genotype as vector of Nloci integers
  #      htfs   : haplotype frequencies
  #      thetas: HWE deviations
  # out: a single probability
  Nloci <- length(gt)
  if (Nloci == 2) {
    if (all(gt %in% c(0, 2))) {
      return(1)
    } else if (any(gt == 0) || any(gt == 2)) {
      return(1/2)
    } else {
      return(htfs[dt[1]] * htfs[dt[2]] * (1 + thetas[dt[1], dt[2]]) /
                (2 * htfs[2] * htfs[3] * (1 + thetas[2, 3]) +
                 2 * htfs[1] * htfs[4] * (1 + thetas[1, 4])   ))
    }
  } else if (Nloci == 3) {
      if (sum(gt == 1) == 0) {
        return(1)
      } else if (sum(gt == 1) == 1) {
        return(1/2)
      } else if (sum(gt == 1) == 2) {
        num <- htfs[dt[1]] * htfs[dt[2]] * (1 + thetas[dt[1], dt[2]])
        if (isTRUE(all.equal(gt, c(0, 1, 1)))) {
          denom <- 2 * htfs[2] * htfs[3] * (1 + thetas[2, 3]) +
                   2 * htfs[1] * htfs[4] * (1 + thetas[1, 4])
        } else if (isTRUE(all.equal(gt, c(1, 0, 1)))) {
          denom <- 2 * htfs[2] * htfs[5] * (1 + thetas[2, 5]) +
                   2 * htfs[1] * htfs[6] * (1 + thetas[1, 6])
        } else if (isTRUE(all.equal(gt, c(1, 1, 0)))) {
          denom <- 2 * htfs[3] * htfs[5] * (1 + thetas[3, 5]) +
                   2 * htfs[1] * htfs[7] * (1 + thetas[1, 7])
        } else if (isTRUE(all.equal(gt, c(1, 1, 2)))) {
          denom <- 2 * htfs[4] * htfs[6] * (1 + thetas[4, 6]) +
                   2 * htfs[2] * htfs[8] * (1 + thetas[2, 8])
        } else if (isTRUE(all.equal(gt, c(1, 2, 1)))) {
          denom <- 2 * htfs[4] * htfs[7] * (1 + thetas[4, 7]) +
                   2 * htfs[3] * htfs[8] * (1 + thetas[3, 8])
        } else if (isTRUE(all.equal(gt, c(2, 1, 1)))) {
          denom <- 2 * htfs[6] * htfs[7] * (1 + thetas[6, 7]) +
                   2 * htfs[5] * htfs[8] * (1 + thetas[5, 8])
        }
        return(num/denom)
      } else {
        num <- htfs[dt[1]] * htfs[dt[2]] * (1 + thetas[dt[1], dt[2]])
        denom <- 2 * htfs[1] * htfs[8] * (1 + thetas[1, 8]) +
                 2 * htfs[2] * htfs[7] * (1 + thetas[2, 7]) +
                 2 * htfs[3] * htfs[6] * (1 + thetas[3, 6]) +
                 2 * htfs[4] * htfs[5] * (1 + thetas[4, 5])
        return(num/denom)
    }
  }
}

subtractHt <- function(gt, ht) {
  # 'Subtracts' given haplotype from a genotype.
  # in:  gt: a genotype as vector of 2 integers
  #      ht: a haplotype as one integer
  # out: the complementary haplotype as integer
  htA <- ord2bin(ht, digits = length(gt))
  if (any((gt == 2 & htA == 0) | (gt == 0 & htA == 1))) {
    return(NA)
  } else {
    return(bin2ord(gt - htA))
  }
}

prob_pi_ik <- function(ht, gt, htfs, thetas = matrix(0, 16, 16)) {
  # Returns probabilities of a haplotype, given genotypes
  # in:  ht    : haplotype as integer
  #      gt    : genotype as vector of Nloci integers (0, 1 or 2)
  #       htfs  : haplotype frequencies
  #       thetas: HWE deviations
  # out: a single probability
  r <- subtractHt(gt, ht)
  if (is.na(r)) {
    return(0)
  } else {
    return(prob_dt_given_gt(c(ht, r), gt, htfs, thetas)) +
           prob_dt_given_gt(c(r, ht), gt, htfs, thetas)
  }
}

expected_lh <- function(gts, htfs, thetas = matrix(0, 16, 16)) {
  # Calculates the expected log-likelihood per individual/row.
  # in:  gts   : genotypes as vectors of Nloci integers (0, 1 or 2)
  #       htfs  : haplotype frequencies
  #       thetas: HWE deviations
  # out: the expected log-likelihood
  N <- nrow(gts)
  Nloci <- ncol(gts)
  M <- 2^Nloci
  elh <- array(0, dim = c(N, M, M))
  for (n in 1:N) {
    gt <- as.numeric(gts[n, ])
    for (htA in 1:M) {
      for (htB in 1:M) {
        if (isTRUE(all.equal(ord2bin(htA, digits = Nloci) +
                             ord2bin(htB, digits = Nloci), gt))) {
          elh[n, htA, htB] <- prob_dt_given_gt(c(htA, htB), gt, htfs, thetas)
          if (!is.finite(elh[n, htA, htB])) {
            return("error")
          } else if (elh[n, htA, htB] < 0) {
            elh[n, htA, htB] <- 0
          }
        }
      }
    }
  }
  return(elh)
}

expected_lh_null <- function(gts, htfs) {
  # Calculates the expected log-likelihood per individual/row.
  # in:  gts   : genotypes as vectors of Nloci integers (0, 1 or 2)
  #       htfs  : haplotype frequencies
  #       thetas: HWE deviations
  # out: the expected log-likelihood
  Nloci <- ncol(gts)
  M <- 2^Nloci
  t(apply(gts, 1, function(gt) {
    sapply(1:M, function(i) {
      return(prob_pi_ik(i, gt, htfs))
    })
  }))
}

dtfsForGts <- function(gts) {
  Nloci <- ncol(gts)
  M <- 2^Nloci
  dtfs <- matrix(NA, M, M)

  if (Nloci == 2) {
    dtfs[1, 1] <- length(gts[gts[, 1] == 0 & gts[, 2] == 0, ])/length(gts)
    dtfs[1, 2] <- length(gts[gts[, 1] == 0 & gts[, 2] == 1, ])/(2*length(gts))
    dtfs[1, 3] <- length(gts[gts[, 1] == 1 & gts[, 2] == 0, ])/(2*length(gts))

    dtfs[2, 2] <- length(gts[gts[, 1] == 0 & gts[, 2] == 2, ])/length(gts)
    dtfs[2, 4] <- length(gts[gts[, 1] == 1 & gts[, 2] == 2, ])/(2*length(gts))

    dtfs[3, 3] <- length(gts[gts[, 1] == 2 & gts[, 2] == 0, ])/length(gts)
    dtfs[3, 4] <- length(gts[gts[, 1] == 2 & gts[, 2] == 1, ])/(2*length(gts))

    dtfs[4, 4] <- length(gts[gts[, 1] == 2 & gts[, 2] == 2, ])/length(gts)
  } else {
    dtfs[1, 1] <- length(gts[gts[, 1] == 0 & gts[, 2] == 0 & gts[, 3] == 0, ])/
                  length(gts)
    dtfs[1, 2] <- length(gts[gts[, 1] == 0 & gts[, 2] == 0 & gts[, 3] == 1, ])/
                  (2*length(gts))
    dtfs[1, 3] <- length(gts[gts[, 1] == 0 & gts[, 2] == 1 & gts[, 3] == 0, ])/
                  (2*length(gts))
    dtfs[1, 5] <- length(gts[gts[, 1] == 1 & gts[, 2] == 0 & gts[, 3] == 0, ])/
                  (2*length(gts))

    dtfs[2, 2] <- length(gts[gts[, 1] == 0 & gts[, 2] == 0 & gts[, 3] == 2, ])/
                  (length(gts))
    dtfs[2, 4] <- length(gts[gts[, 1] == 0 & gts[, 2] == 1 & gts[, 3] == 2, ])/
                  (2*length(gts))
    dtfs[2, 6] <- length(gts[gts[, 1] == 1 & gts[, 2] == 0 & gts[, 3] == 2, ])/
                  (2*length(gts))

    dtfs[3, 3] <- length(gts[gts[, 1] == 0 & gts[, 2] == 2 & gts[, 3] == 0, ])/
                  (length(gts))
    dtfs[3, 4] <- length(gts[gts[, 1] == 0 & gts[, 2] == 2 & gts[, 3] == 1, ])/
                  (2*length(gts))
    dtfs[3, 7] <- length(gts[gts[, 1] == 1 & gts[, 2] == 2 & gts[, 3] == 0, ])/
                  (2*length(gts))

    dtfs[4, 4] <- length(gts[gts[, 1] == 0 & gts[, 2] == 2 & gts[, 3] == 2, ])/
```

```
                  (length(gts))
    dtfs[4, 8] <- length(gts[gts[, 1] == 1 & gts[, 2] == 2 & gts[, 3] == 2, ])/
                  (2*length(gts))

    dtfs[5, 5] <- length(gts[gts[, 1] == 2 & gts[, 2] == 0 & gts[, 3] == 0, ])/
                  (length(gts))
    dtfs[5, 6] <- length(gts[gts[, 1] == 2 & gts[, 2] == 0 & gts[, 3] == 1, ])/
                  (2*length(gts))
    dtfs[5, 7] <- length(gts[gts[, 1] == 2 & gts[, 2] == 1 & gts[, 3] == 0, ])/
                  (2*length(gts))

    dtfs[6, 6] <- length(gts[gts[, 1] == 2 & gts[, 2] == 0 & gts[, 3] == 2, ])/
                  (length(gts))
    dtfs[6, 8] <- length(gts[gts[, 1] == 2 & gts[, 2] == 1 & gts[, 3] == 2, ])/
                  (2*length(gts))

    dtfs[7, 7] <- length(gts[gts[, 1] == 2 & gts[, 2] == 2 & gts[, 3] == 0, ])/
                  (length(gts))
    dtfs[7, 8] <- length(gts[gts[, 1] == 2 & gts[, 2] == 2 & gts[, 3] == 1, ])/
                  (2*length(gts))

    dtfs[8, 8] <- length(gts[gts[, 1] == 2 & gts[, 2] == 2 & gts[, 3] == 2, ])/
                  (length(gts))
  }

  dtfs[lower.tri(dtfs)] <- t(dtfs)[lower.tri(dtfs)]
  return(dtfs)
}

calcThetasE <- function(gts, htfsE, method, alpha = NA, dtfsE = NA,
                        thetas0 = NA) {
  Nloci <- ncol(gts)
  M <- 2^ncol(gts)
  thetasE <- matrix(0, M, M)
  if (method == "deltas_equal") {
    if (Nloci == 2) {
      dtfs <- dtfsForGts(gts)
      dtfs[1, 4] <- length(gts[gts[, 1] == 1 & gts[, 2] == 1, ])/(4*length(gts))
      dtfs[2, 3] <- dtfs[3, 2] <- dtfs[4, 1] <- dtfs[1, 4]
    } else if (Nloci == 3) {
      dtfs <- dtfsForGts(gts)
      dtfs[1, 4] <- length(gts[gts[, 1] == 0 & gts[, 2] == 1 & gts[, 3] == 1, ])/
                    (4*length(gts))
      dtfs[1, 6] <- length(gts[gts[, 1] == 1 & gts[, 2] == 0 & gts[, 3] == 1, ])/
                    (4*length(gts))
      dtfs[1, 7] <- length(gts[gts[, 1] == 1 & gts[, 2] == 1 & gts[, 3] == 0, ])/
                    (4*length(gts))
      dtfs[2, 8] <- length(gts[gts[, 1] == 1 & gts[, 2] == 1 & gts[, 3] == 2, ])/
                    (4*length(gts))
      dtfs[3, 8] <- length(gts[gts[, 1] == 1 & gts[, 2] == 2 & gts[, 3] == 1, ])/
                    (4*length(gts))
      dtfs[5, 8] <- length(gts[gts[, 1] == 2 & gts[, 2] == 1 & gts[, 3] == 1, ])/
                    (4*length(gts))

      dtfs[1, 8] <- length(gts[gts[, 1] == 1 & gts[, 2] == 1 & gts[, 3] == 1, ])/
                    (8*length(gts))

      dtfs[2, 3] <- dtfs[1, 4]
      dtfs[2, 5] <- dtfs[1, 6]
      dtfs[3, 5] <- dtfs[1, 7]
      dtfs[4, 6] <- dtfs[2, 8]
      dtfs[4, 7] <- dtfs[3, 8]
      dtfs[6, 7] <- dtfs[5, 8]
      dtfs[2, 7] <- dtfs[3, 6] <- dtfs[4, 5] <- dtfs[1, 8]
      dtfs[lower.tri(dtfs)] <- t(dtfs)[lower.tri(dtfs)]
    }
    for (i in 1:M) {
      for (j in 1:M) {
        thetasE[i, j] <- dtfs[i, j] / (htfsE[i] * htfsE[j]) - 1
      }
    }
    return(thetasE)
  } else if (method == "HWE") {
    dtfs <- dtfsForGts(gts)
    for (i in 1:M) {
      for (j in 1:M) {
        if (is.na(dtfs[i, j])) {
          thetasE[i, j] <- 0
        } else {
          thetasE[i, j] <- dtfs[i, j] / (htfsE[i] * htfsE[j]) - 1
        }
      }
    }
    return(thetasE)
  }
}

updateParameters_fixed <- function(thetaFixed) {
  function(gts, htfsE, dtfsE) {
    return(list(eta = htfsE, theta = thetaFixed))
  }
}

updateParameters03_fixed <- function(gts, htfsE, dtfsE, theta03) {
  theta <- calcThetasE(gts, htfsE, method = "HWE")
  pi_11 <- length(gts[gts[, 1] == 1 & gts[, 2] == 1, ])/length(gts)
  theta[1, 4] <- theta[4, 1] <- theta03
  theta[2, 3] <- theta[3, 2] <- (0.5*pi_11 - htfsE[1] * htfsE[4] *
                                (1 + theta03)) / (htfsE[2] * htfsE[3]) - 1
  if (any(!is.finite(theta))) {
    theta[!is.finite(theta)] <- 0
  }
  return(list(eta = htfsE, theta = theta))
}

updateParameters_deltas_equal <- function(gts, htfsE, dtfsE) {
  return(list(eta = htfsE, theta = calcThetasE(gts, htfsE,
                                               method = "deltas_equal")))
}

updateParameters_hwe <- function(gts, htfsE, dtfsE) {
  return(list(eta = htfsE, theta = calcThetasE(gts, htfsE, method = "HWE")))
}

updateThetaStrat = function(eta, theta, alpha, gts) {
  dtfsE <- (theta + 1) * deltaExp(eta)
  parsConst <- list(alpha = alpha, eta = eta, deltaE = dtfsE, gts = gts)
  eta1 <- optim(eta, parEst, method = 'Nelder-Mead', parsConst = parsConst)$par
  # print(eta1)
  eta2 <- etaOther(eta1, eta, alpha)
  # print(eta2)
  theta03E <- (alpha * eta1[1] * eta1[4] + (1-alpha) * eta2[1] * eta2[4]) /
              ((alpha * eta1[1] + (1-alpha) * eta2[1]) *
               (alpha * eta1[4] + (1-alpha) * eta2[4])) - 1
  return(list(theta03E = theta03E, htfs1 = eta1, htfs2 = eta2))
}

vn <- function(v)v/sum(v)

etaMixed <- function(eta1, eta2, alpha) {
  (alpha * eta1 + (1 - alpha) * eta2)
}

etaOther <- function(eta1, eta, alpha) {
  (eta - alpha * eta1) / (1 - alpha)
}

deltaExp <- function(eta) {
  matrix(kronecker(eta, eta), ncol = length(eta))
}

deltaMixed <- etaMixed

# pars: list with alpha, eta, theta (matrix)
# eta12: linear parameter vectors with etas for pop1, pop2
parEst <- function(eta1, parsConst) with(parsConst, {
  if (any(eta1 < 0 | eta1 > 1)) {
    return(Inf)
  }

  eta2 <- etaOther(eta1, eta, alpha)

  if (any(eta2 < 0 | eta2 > 1)) {
    return(Inf)
  }

  deltaE1 <- deltaExp(eta1)
  deltaE2 <- deltaExp(eta2)
  deltaO <- deltaMixed(deltaE1, deltaE2, alpha)

  C1 <- deltaE - deltaO

  # translate to optimization goal
  R <- sum(abs(C1) / deltaE)

  return(R)
})

haplotype_em_generic <- function(gts, eps = 1e-5, updateParameters, ...) {
  # Implements the EM-algorithm to estimate haplotype frequencies.
  # in:  gts: genotypes as vectors of 2 integers (0, 1 or 2)
  # out: estimated haplotype frequencies
  M <- 2^ncol(gts)

  htfs0 <- htfsE <- rep(1/M, M)
  thetas0 <- thetasE <- matrix(0, M, M)

  repeat {
    # E-step
    elh <- expected_lh(gts, htfs0, thetas0)
    if (any(elh == "error")) {
      return("error")
    }
    # M-step
    dtfsE <- apply(elh, c(2, 3), mean)
    htfsE <- rowSums(dtfsE)
    newPars <- updateParameters(gts, htfsE, dtfsE, ...)
    thetasE <- newPars$theta
    htfsE <- newPars$eta

    if (any(thetasE <= -1)) {
      return("error")
    }
    if (all(abs(htfs0 - htfsE) < eps) & all(abs(thetas0 - thetasE) < eps)) {
      break;
    } else {
      htfs0 <- htfsE
      thetas0 <- thetasE
    }
  }
  return(list(htfs = htfsE, thetas = thetasE))
}

haplotype_em_strat <- function(gts, alpha = 0.5, eps = 1e-5) {
  theta030s <- c(0, 0.1, -0.1, 0.5, -0.5)
  for (i in 1:length(theta030s)) {
    theta030 <- theta030s[i]
    theta03E <- 0
    newPars <- list()
    error <- 0
    ct <- 0
    repeat {
      newPars <- haplotype_em_generic(gts, updateParameters = updateParameters03_
                                      theta03 = theta030)
      if (any(newPars == "error")) {
        error <- 1
        break;
      }
      new <- updateThetaStrat(newPars$htfs, newPars$thetas, alpha, gts)
      theta03E <- new$theta03E
      ct <- ct + 1
      if (ct > 15) {
        error <- 2
        break;
      }
      if (abs(theta03E - theta030) < eps) {
        break;
      } else {
        theta030 <- theta03E
      }
    }
    if (error == 0) {
      break;
    }
  }
  if (any(newPars == "error")) {
    return(newPars)
  } else {
    return(append(newPars, list(htfs1 = new$htfs1/sum(new$htfs1),
                                htfs2 = new$htfs2/sum(new$htfs2))))
  }
}

haplotype_em_null <- function(gts, eps = 1e-3) {
  # Implements the EM-algorithm to estimate haplotype frequencies.
  # in:  gts: genotypes as vectors of 2 integers (0, 1 or 2)
  # out: estimated haplotype frequencies
  M <- 2^ncol(gts)

  htfs0 <- rep(1/M, M)
  htfsE <- NULL
  repeat {
    # E-step
    elh <- expected_lh_null(gts, htfs0)
    # M-step
    htfsE <- colSums(elh) / sum(elh)
    if (all(abs(htfs0 - htfsE) < eps)) {
      break;
    } else {
      htfs0 <- htfsE
    }
  }
  return(htfs = htfsE)
}
```

B.3 Running simulations

```
runSimulation <- function(N, Nloci, alpha = NA, beta = 1e3, HWE = F, strat = F,
                          updateParameters, ...) {
  Nreps <- 500
  M <- 2 ^ Nloci

  htfs_data <- matrix(NA, Nreps, M)
  htfs_est0 <- matrix(NA, Nreps, M)
  htfs_est1 <- matrix(NA, Nreps, M)
  htfs_est2 <- matrix(NA, Nreps, M)

  thetas_data <- array(NA, c(M, M, Nreps))
  thetas_est1 <- array(NA, c(M, M, Nreps))
  thetas_est2 <- array(NA, c(M, M, Nreps))

  if (strat == F) {
    results <- data.frame(MSE_htfs0 = numeric(),
                          MSE_htfs1 = numeric(),
                          MSE_thetas1 = numeric(),
                          MSE_htfs2 = numeric(),
                          MSE_thetas2 = numeric())
  } else {
    results <- data.frame(MSE_htfs0 = numeric(),
                          MSE_htfs1 = numeric(),
                          MSE_thetas1 = numeric(),
                          MSE_htfs2 = numeric(),
                          MSE_thetas2 = numeric(),
                          MSE_htfssub1 = numeric(),
                            MSE_htfssub2 = numeric())
  }

  for (i in 1:Nreps) {
    data <- simGts(N, Nloci = Nloci, HWE = HWE, strat = strat, alpha = alpha[1],
                   beta = beta)
    out0 <- haplotype_em_null(data$gts)
    out1 <- haplotype_em_generic(data$gts,
            updateParameters = updateParameters_fixed(thetaFixed = data$thetas),
            ...)
    if (isTRUE(is.na(alpha))) {
      out2 <- haplotype_em_generic(data$gts,
                                   updateParameters = updateParameters,
                                   ...)
    } else {
      out2 <- haplotype_em_strat(data$gts, alpha = alpha[2])
      if (any(out2 == "error")) {
        next;
      }
    }

    htfs_data[i, ] <- data$htfs
    htfs_est0[i, ] <- out0
    htfs_est1[i, ] <- out1$htfs
    htfs_est2[i, ] <- out2$htfs

    thetas_data[, , i] <- data$thetas
    thetas_est1[, , i] <- out1$thetas
    thetas_est2[, , i] <- out2$thetas

    results[i, "MSE_htfs0"] <- mean((data$htfs - out0)^2)
    results[i, "MSE_htfs1"] <- mean((data$htfs - out1$htfs)^2)
    results[i, "MSE_thetas1"] <- mean((data$thetas - out1$thetas)^2)

    results[i, "MSE_htfs2"] <- mean((data$htfs - out2$htfs)^2)
    results[i, "MSE_thetas2"] <- mean((data$thetas - out2$thetas)^2)

    if (strat == T) {
      results[i, "MSE_htfssub1"] <- mean((data$htfs1 - out2$htfs1)^2)
      results[i, "MSE_htfssub2"] <- mean((data$htfs2 - out2$htfs2)^2)
    }
    print(c("i=",i))
  }
  return(list(htfs_data = htfs_data, htfs_est0 = htfs_est0,
              htfs_est1 = htfs_est1, htfs_est2 = htfs_est2,
              thetas_data = thetas_data, thetas_est1 = thetas_est1,
              thetas_est2 = thetas_est2,
              results = results))
}
```

B.4 Data analysis

```
doAnalysis <- function(data) {
  snps <- colnames(data)[3:ncol(data)]
  pops <- c("YRI", "CEU", "CHB", "JPT")

  df <- data.frame(snp1 = character(),
                   snp2 = character(),
                   h0 = numeric(),
                   h1 = numeric(),
                   h2 = numeric(),
                   h3 = numeric(),
                   stringsAsFactors = FALSE)

  htfs0 <- list(YRI = df,
                CEU = df,
                CHB = df,
                JPT = df)

  htfs1 <- data.frame(snp1 = character(),
                      snp2 = character(),
                      h0 = numeric(),
                      h1 = numeric(),
                      h2 = numeric(),
                      h3 = numeric(),
                      h0_1 = numeric(),
                      h1_1 = numeric(),
                      h2_1 = numeric(),
                      h3_1 = numeric(),
                      h0_2 = numeric(),
                      h1_2 = numeric(),
                      h2_2 = numeric(),
                      h3_2 = numeric(),
                      stringsAsFactors = FALSE)

  for (i in 1:(length(snps) - 1)) {
    snpsub <- snps[c(i, i+1)]
    datasub <- data[, c("id", "pop", snpsub)]
    datasub <- datasub[complete.cases(datasub), ]

    # Subpopulations separately
    for (p in 1:4) {
      if (nrow(datasub[datasub$pop == pops[p], 3:4]) >= 20) {
        htfs <- haplotype_em_null(datasub[datasub$pop == pops[p], 3:4])
        htfs0[[p]][i, 1:2] <- snpsub
        htfs0[[p]][i, 3:6] <- round(htfs, 4)
      } else {
        htfs0[[p]][i, 1:2] <- snpsub
        next
      }
    }

    if (isTRUE(all((htfs0$YRI[i, 3:6] + htfs0$CEU[i, 3:6]) > 0))) {
      tryCatch({
        datasub <- data[data$pop %in% c("CEU", "YRI"), c("id", "pop", snpsub)]
        datasub <- datasub[complete.cases(datasub), ]
        out <- haplotype_em_strat(datasub[, 3:4], alpha = 0.5)
        htfs1[i, 1:2] <- snpsub
        htfs1[i, 3:6] <- out$htfs
        htfs1[i, 7:10] <- out$htfs1
        htfs1[i, 11:14] <- out$htfs2
      }, error = function(e){
        htfs1[i, 1:2] <- snpsub
        htfs1[i, 3:14] <- NA})
    }
    print(c("i=", i))
  }
  return(list(htfs0 = htfs0, htfs1 = htfs1))
}
```

Appendix C

Haplotype frequencies SNP pairs by population

C.1 Chromosome 6: HLA-A gene

Table C.1: Haplotype frequencies HLA-A region, population YRI

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs2523793	rs2735020	.0773	.5280	.0017	.3931
rs2735020	rs2517889	.0744	.0015	.5268	.3972
rs2517889	rs2523790	.6250	.0000	.0000	.3750
rs2523790	rs16896049	.0000	.6190	.0000	.3810
rs16896049	rs16896051	.0000	.0000	.2753	.7247
rs16896051	rs1233326	.0007	.2944	.0836	.6212
rs1233326	rs9404952	.0852	.0002	.2990	.6156
rs9404952	rs16896052	.0000	.3989	.0000	.6011
rs16896052	rs2523786	.0000	.0000	.6236	.3764
rs2523786	rs2394179	.6236	.0000	.0000	.3764
rs2394179	rs2394180	.5055	.1167	.0001	.3777
rs2394180	rs2523783	.0007	.5049	.1160	.3785
rs2523783	rs2394182	.1150	.0013	.5071	.3766
rs2394182	rs2735015	.6296	.0062	.0000	.3642
rs2735015	rs2735014	.6312	.0000	.0063	.3625
rs2735014	rs9468641	.0351	.5884	.0002	.3763
rs9468641	rs2253981	.0388	.0001	.5835	.3776
rs2253981	rs7755504	.2219	.4004	.0004	.3774
rs7755504	rs2254071	.2219	.0004	.4004	.3774
rs2254071	rs1233324	.0001	.6463	.0792	.2745
rs1233324	rs2254077	.0001	.0843	.6493	.2663
rs2254077	rs2523781	.1206	.5013	.0013	.3767
rs2523781	rs2523780	.1124	.0007	.5126	.3743
rs2523780	rs2523779	.1111	.5124	.0007	.3758
rs2523779	rs9258525	NA	NA	NA	NA

Table C.2: Haplotype frequencies HLA-A region, population CEU

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs2523793	rs2735020	.0343	.1598	.0068	.7990
rs2735020	rs2517889	.0354	.0073	.1842	.7732
rs2517889	rs2523790	.2442	.0000	.0000	.7558
rs2523790	rs16896049	.0332	.2168	.0001	.7499
rs16896049	rs16896051	.0000	.0333	.1056	.8611
rs16896051	rs1233326	.0007	.1048	.0715	.8229
rs1233326	rs9404952	.0717	.0005	.4339	.4939
rs9404952	rs16896052	.0344	.4713	.0001	.4942
rs16896052	rs2523786	.0352	.0001	.2118	.7529
rs2523786	rs2394179	.2321	.0000	.0000	.7679
rs2394179	rs2394180	.1117	.1236	.0000	.7647
rs2394180	rs2523783	.0017	.1275	.0825	.7883
rs2523783	rs2394182	.0986	.0000	.1777	.7236
rs2394182	rs2735015	.2661	.0000	.0000	.7339
rs2735015	rs2735014	.1805	.0417	.0000	.7778
rs2735014	rs9468641	.0000	.2135	.0000	.7865
rs9468641	rs2253981	.0000	.0000	.2500	.7500
rs2253981	rs7755504	.0168	.2304	.0001	.7527
rs7755504	rs2254071	.0168	.0001	.2304	.7527
rs2254071	rs1233324	.0007	.2579	.0568	.6846
rs1233324	rs2254077	.0007	.0568	.2579	.6846
rs2254077	rs2523781	.0909	.1689	.0000	.7402
rs2523781	rs2523780	.0941	.0001	.1378	.7680
rs2523780	rs2523779	.0885	.1330	.0001	.7784
rs2523779	rs9258525	.0057	.0805	.0690	.8448

Table C.3: Haplotype frequencies HLA-A region, population CHB

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs2523793	rs2735020	.0118	.2443	.0004	.7435
rs2735020	rs2517889	.0069	.0053	.4322	.5556
rs2517889	rs2523790	.4545	.0000	.0000	.5455
rs2523790	rs16896049	.1666	.2889	.0000	.5444
rs16896049	rs16896051	.0010	.1657	.1546	.6788
rs16896051	rs1233326	.0061	.1416	.0280	.8243
rs1233326	rs9404952	.0333	.0008	.3303	.6356
rs9404952	rs16896052	.1703	.2047	.0002	.6248
rs16896052	rs2523786	.1628	.0000	.2791	.5581
rs2523786	rs2394179	.4432	.0000	.0000	.5568
rs2394179	rs2394180	.2444	.2112	.0001	.5444
rs2394180	rs2523783	.0000	.2444	.0444	.7111
rs2523783	rs2394182	.0304	.0054	.4339	.5304

rs2394182	rs2735015	.4744	.0000	.0000	.5256
rs2735015	rs2735014	.2976	.1667	.0001	.5356
rs2735014	rs9468641	.0000	.2889	.0000	.7111
rs9468641	rs2253981	.0000	.0000	.4556	.5444
rs2253981	rs7755504	.0888	.3667	.0001	.5444
rs7755504	rs2254071	.0888	.0001	.3667	.5444
rs2254071	rs1233324	.0010	.4545	.0434	.5010
rs1233324	rs2254077	.0010	.0434	.4545	.5010
rs2254077	rs2523781	.0411	.4144	.0033	.5411
rs2523781	rs2523780	.0423	.0032	.4123	.5423
rs2523780	rs2523779	.0420	.4231	.0045	.5304
rs2523779	rs9258525	NA	NA	NA	NA

Table C.4: Haplotype frequencies HLA-A region, population JPT

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs2523793	rs2735020	.0000	.5000	.0000	.5000
rs2735020	rs2517889	.0000	.0000	.5125	.4875
rs2517889	rs2523790	.5000	.0000	.0000	.5000
rs2523790	rs16896049	.0058	.4942	.0058	.4942
rs16896049	rs16896051	.0000	.0116	.3488	.6395
rs16896051	rs1233326	.0000	.3372	.0116	.6512
rs1233326	rs9404952	.0116	.0000	.2442	.7442
rs9404952	rs16896052	.0114	.2273	.0000	.7614
rs16896052	rs2523786	.0056	.0058	.5058	.4828
rs2523786	rs2394179	.5111	.0000	.0000	.4889
rs2394179	rs2394180	.4767	.0233	.0000	.5000
rs2394180	rs2523783	.0000	.4767	.0000	.5233
rs2523783	rs2394182	.0000	.0000	.5000	.5000
rs2394182	rs2735015	.4875	.0000	.0000	.5125
rs2735015	rs2735014	.4625	.0250	.0000	.5125
rs2735014	rs9468641	.0000	.4767	.0000	.5233
rs9468641	rs2253981	.0000	.0000	.5111	.4889
rs2253981	rs7755504	.1332	.3779	.0001	.4888
rs7755504	rs2254071	.1332	.0001	.3779	.4888
rs2254071	rs1233324	.0000	.5111	.0111	.4778
rs1233324	rs2254077	.0000	.0114	.5000	.4886
rs2254077	rs2523781	.0000	.5000	.0000	.5000
rs2523781	rs2523780	.0000	.0000	.5119	.4881
rs2523780	rs2523779	.0000	.5119	.0000	.4881
rs2523779	rs9258525	NA	NA	NA	NA

C.2 Chromosome 7: HOX-A10 gene

Table C.5: Haplotype frequencies HOX-A10 region, population YRI

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs17427984	rs17427991	.0000	.0000	.0000	1.000
rs17427991	rs28357156	.0000	.0000	.0000	1.000
rs28357156	rs4722675	.0000	.0000	.0000	1.000
rs4722675	rs17501326	NA	NA	NA	NA
rs17501326	rs929250	NA	NA	NA	NA
rs929250	rs3735533	.0000	.0000	.0000	1.000
rs3735533	rs10228276	.0000	.0000	.4593	.5407
rs10228276	rs17472490	.0002	.4686	.1748	.3564
rs17472490	rs28357160	.0000	.1867	.0000	.8133
rs28357160	rs12671338	.0000	.0000	.0000	1.000
rs12671338	rs12671340	.0000	.0000	.0000	1.000
rs12671340	rs17472497	.0000	.0000	.0000	1.000
rs17472497	rs28357161	.0000	.0000	.0000	1.000
rs28357161	rs13243033	.0000	.0000	.6742	.3258
rs13243033	rs17501347	.1575	.5274	.0137	.3014
rs17501347	rs17437636	.0000	.1667	.0000	.8333
rs17437636	rs17501354	.0000	.0000	.0471	.9529
rs17501354	rs7812039	.0003	.0457	.6721	.2819
rs7812039	rs17449303	.0000	.6706	.0059	.3235
rs17449303	rs11973735	NA	NA	NA	NA
rs11973735	rs28357162	NA	NA	NA	NA
rs28357162	rs17437657	.0051	.0449	.0199	.9301
rs17437657	rs28357163	.0000	.0250	.0000	.9750
rs28357163	rs17501375	NA	NA	NA	NA
rs17501375	rs17501382	NA	NA	NA	NA
rs17501382	rs17501389	NA	NA	NA	NA
rs17501389	rs17428012	NA	NA	NA	NA
rs17428012	rs28357164	.0000	.1404	.0000	.8596
rs28357164	rs7786570	.0000	.0000	.0795	.9205
rs7786570	rs17501403	.0000	.0833	.0357	.8810
rs17501403	rs17428025	.0057	.0296	.1708	.7939
rs17428025	rs17428032	.0061	.1798	.0003	.8138
rs17428032	rs17472574	.0000	.0068	.0000	.9932
rs17472574	rs6958837	.0000	.0000	.5641	.4359
rs6958837	rs28357165	.0000	.5536	.0000	.4464
rs28357165	rs17437670	.0000	.0000	.0238	.9762
rs17437670	rs17472588	.0000	.0238	.1250	.8512
rs17472588	rs4722676	NA	NA	NA	NA
rs4722676	rs17501340	NA	NA	NA	NA
rs17501340	rs17472504	NA	NA	NA	NA
rs17472504	rs17501361	NA	NA	NA	NA
rs17501361	rs17501396	NA	NA	NA	NA

rs17501396	rs17472560	NA	NA	NA	NA
rs17472560	rs17501424	NA	NA	NA	NA
rs17501424	rs17501368	NA	NA	NA	NA
rs17501368	rs17472532	.0000	.1688	.0188	.8125
rs17472532	rs17428018	.0000	.0211	.0000	.9789
rs17428018	rs10248288	.0000	.0000	.0169	.9831
rs10248288	rs17437677	.0000	.0169	.0000	.9831

Table C.6: Haplotype frequencies HOX-A10 region, population CEU

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs17427984	rs17427991	.0000	.0000	.0000	1.000
rs17427991	rs28357156	.0000	.0000	.0000	1.000
rs28357156	rs4722675	.0000	.0000	.0455	.9545
rs4722675	rs17501326	.0000	.0366	.0305	.9329
rs17501326	rs929250	.0000	.0301	.0422	.9277
rs929250	rs3735533	.0500	.0000	.0000	.9500
rs3735533	rs10228276	.0528	.0001	.1413	.8058
rs10228276	rs17472490	.0000	.1975	.0000	.8025
rs17472490	rs28357160	.0000	.0000	.0000	1.000
rs28357160	rs12671338	.0000	.0000	.0000	1.000
rs12671338	rs12671340	.0000	.0000	.0000	1.000
rs12671340	rs17472497	NA	NA	NA	NA
rs17472497	rs28357161	NA	NA	NA	NA
rs28357161	rs13243033	.0000	.0000	.1854	.8146
rs13243033	rs17501347	NA	NA	NA	NA
rs17501347	rs17437636	NA	NA	NA	NA
rs17437636	rs17501354	NA	NA	NA	NA
rs17501354	rs7812039	NA	NA	NA	NA
rs7812039	rs17449303	.0041	.1988	.0104	.7867
rs17449303	rs11973735	.0056	.0075	.0931	.8938
rs11973735	rs28357162	.0000	.0952	.0119	.8929
rs28357162	rs17437657	.0000	.0112	.1067	.8820
rs17437657	rs28357163	.0000	.1067	.0000	.8933
rs28357163	rs17501375	.0000	.0000	.0195	.9805
rs17501375	rs17501382	.0000	.0205	.0000	.9795
rs17501382	rs17501389	NA	NA	NA	NA
rs17501389	rs17428012	NA	NA	NA	NA
rs17428012	rs28357164	.0000	.1067	.0000	.8933
rs28357164	rs7786570	.0000	.0000	.0000	1.000
rs7786570	rs17501403	NA	NA	NA	NA
rs17501403	rs17428025	NA	NA	NA	NA
rs17428025	rs17428032	NA	NA	NA	NA
rs17428032	rs17472574	NA	NA	NA	NA
rs17472574	rs6958837	.0000	.0000	.0814	.9186
rs6958837	rs28357165	.0000	.0814	.0000	.9186
rs28357165	rs17437670	.0000	.0000	.0750	.9250

rs17437670	rs17472588	NA	NA	NA	NA
rs17472588	rs4722676	NA	NA	NA	NA
rs4722676	rs17501340	.0000	.0000	.0000	1.000
rs17501340	rs17472504	.0000	.0000	.0000	1.000
rs17472504	rs17501361	.0000	.0000	.0000	1.000
rs17501361	rs17501396	.0000	.0000	.0000	1.000
rs17501396	rs17472560	.0000	.0000	.0000	1.000
rs17472560	rs17501424	.0000	.0000	.0000	1.000
rs17501424	rs17501368	NA	NA	NA	NA
rs17501368	rs17472532	NA	NA	NA	NA
rs17472532	rs17428018	NA	NA	NA	NA
rs17428018	rs10248288	NA	NA	NA	NA
rs10248288	rs17437677	NA	NA	NA	NA

Table C.7: Haplotype frequencies HOX-A10 region, population CHB

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs17427984	rs17427991	.0000	.0000	.0000	1.000
rs17427991	rs28357156	.0000	.0000	.0227	.9773
rs28357156	rs4722675	.0000	.0227	.3068	.6705
rs4722675	rs17501326	.0000	.3068	.0000	.6932
rs17501326	rs929250	.0000	.0000	.2907	.7093
rs929250	rs3735533	.2727	.0114	.0000	.7159
rs3735533	rs10228276	.2889	.0000	.0111	.7000
rs10228276	rs17472490	.0000	.2791	.0000	.7209
rs17472490	rs28357160	.0000	.0000	.0000	1.000
rs28357160	rs12671338	.0000	.0000	.0000	1.000
rs12671338	rs12671340	.0000	.0000	.0000	1.000
rs12671340	rs17472497	.0000	.0000	.0000	1.000
rs17472497	rs28357161	.0000	.0000	.0000	1.000
rs28357161	rs13243033	.0000	.0000	.3023	.6977
rs13243033	rs17501347	.0000	.3000	.0000	.7000
rs17501347	rs17437636	.0000	.0000	.0000	1.000
rs17437636	rs17501354	.0000	.0000	.0000	1.000
rs17501354	rs7812039	.0000	.0000	.3182	.6818
rs7812039	rs17449303	.0125	.2824	.0003	.7048
rs17449303	rs11973735	.0000	.0132	.0000	.9868
rs11973735	rs28357162	.0000	.0000	.0114	.9886
rs28357162	rs17437657	.0000	.0116	.0000	.9884
rs17437657	rs28357163	.0000	.0000	.0000	1.000
rs28357163	rs17501375	.0000	.0000	.0227	.9773
rs17501375	rs17501382	.0000	.0244	.0000	.9756
rs17501382	rs17501389	.0000	.0000	.0000	1.000
rs17501389	rs17428012	.0000	.0000	.0000	1.000
rs17428012	rs28357164	.0000	.0000	.0000	1.000
rs28357164	rs7786570	.0000	.0000	.0000	1.000
rs7786570	rs17501403	.0000	.0000	.0000	1.000

rs17501403	rs17428025	.0000	.0000	.0000	1.000
rs17428025	rs17428032	.0000	.0000	.0000	1.000
rs17428032	rs17472574	.0000	.0000	.0000	1.000
rs17472574	rs6958837	.0000	.0000	.4024	.5976
rs6958837	rs28357165	.0064	.4158	.0047	.5731
rs28357165	rs17437670	.0000	.0119	.0000	.9881
rs17437670	rs17472588	.0000	.0000	.0000	1.000
rs17472588	rs4722676	NA	NA	NA	NA
rs4722676	rs17501340	NA	NA	NA	NA
rs17501340	rs17472504	NA	NA	NA	NA
rs17472504	rs17501361	NA	NA	NA	NA
rs17501361	rs17501396	NA	NA	NA	NA
rs17501396	rs17472560	NA	NA	NA	NA
rs17472560	rs17501424	NA	NA	NA	NA
rs17501424	rs17501368	NA	NA	NA	NA
rs17501368	rs17472532	NA	NA	NA	NA
rs17472532	rs17428018	NA	NA	NA	NA
rs17428018	rs10248288	NA	NA	NA	NA
rs10248288	rs17437677	NA	NA	NA	NA

Table C.8: Haplotype frequencies HOX-A10 region, population JPT

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs17427984	rs17427991	.0000	.0000	.0000	1.000
rs17427991	rs28357156	.0000	.0000	.0116	.9884
rs28357156	rs4722675	.0000	.0116	.4070	.5814
rs4722675	rs17501326	.0000	.3846	.0000	.6154
rs17501326	rs929250	.0000	.0000	.4250	.5750
rs929250	rs3735533	.4432	.0000	.0000	.5568
rs3735533	rs10228276	.4419	.0116	.0000	.5465
rs10228276	rs17472490	.0000	.4390	.0000	.5610
rs17472490	rs28357160	.0000	.0000	.0122	.9878
rs28357160	rs12671338	.0000	.0114	.0000	.9886
rs12671338	rs12671340	.0000	.0000	.0000	1.000
rs12671340	rs17472497	.0000	.0000	.0135	.9865
rs17472497	rs28357161	.0000	.0125	.0000	.9875
rs28357161	rs13243033	.0000	.0000	.4306	.5694
rs13243033	rs17501347	.0000	.4516	.0000	.5484
rs17501347	rs17437636	.0000	.0000	.0000	1.000
rs17437636	rs17501354	.0000	.0000	.0000	1.000
rs17501354	rs7812039	.0000	.0000	.4146	.5854
rs7812039	rs17449303	.0000	.4268	.0000	.5732
rs17449303	rs11973735	.0000	.0000	.0000	1.000
rs11973735	rs28357162	.0000	.0000	.0341	.9659
rs28357162	rs17437657	.0000	.0349	.0000	.9651
rs17437657	rs28357163	.0000	.0000	.0000	1.000
rs28357163	rs17501375	.0000	.0000	.0000	1.000

rs17501375	rs17501382	.0000	.0000	.0000	1.000
rs17501382	rs17501389	.0000	.0000	.0000	1.000
rs17501389	rs17428012	.0000	.0000	.0000	1.000
rs17428012	rs28357164	.0000	.0000	.0114	.9886
rs28357164	rs7786570	.0000	.0116	.0000	.9884
rs7786570	rs17501403	.0000	.0000	.0000	1.000
rs17501403	rs17428025	.0000	.0000	.0000	1.000
rs17428025	rs17428032	.0000	.0000	.0000	1.000
rs17428032	rs17472574	.0000	.0000	.0000	1.000
rs17472574	rs6958837	.0000	.0000	.4865	.5135
rs6958837	rs28357165	.0116	.4419	.0000	.5465
rs28357165	rs17437670	.0000	.0125	.0000	.9875
rs17437670	rs17472588	.0000	.0000	.0000	1.000
rs17472588	rs4722676	NA	NA	NA	NA
rs4722676	rs17501340	NA	NA	NA	NA
rs17501340	rs17472504	NA	NA	NA	NA
rs17472504	rs17501361	NA	NA	NA	NA
rs17501361	rs17501396	NA	NA	NA	NA
rs17501396	rs17472560	NA	NA	NA	NA
rs17472560	rs17501424	NA	NA	NA	NA
rs17501424	rs17501368	NA	NA	NA	NA
rs17501368	rs17472532	NA	NA	NA	NA
rs17472532	rs17428018	NA	NA	NA	NA
rs17428018	rs10248288	NA	NA	NA	NA
rs10248288	rs17437677	NA	NA	NA	NA

C.3 Chromosome 8: gene desert

Table C.9: Haplotype frequencies gene desert, population YRI

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs4909620	rs7357354	NA	NA	NA	NA
rs7357354	rs12677926	NA	NA	NA	NA
rs12677926	rs4909621	.0000	.0000	.4940	.5060
rs4909621	rs7008788	.4944	.0000	.0000	.5056
rs7008788	rs6995856	.0015	.5102	.0160	.4724
rs6995856	rs11777213	.0190	.0000	.0063	.9747
rs11777213	rs11776500	.0422	.0060	.0000	.9518
rs11776500	rs11786769	.0006	.0383	.4994	.4617
rs11786769	rs17607388	.0198	.4802	.0025	.4975
rs17607388	rs10086749	.0037	.0186	.4630	.5148
rs10086749	rs12547899	.0005	.4662	.0273	.5060
rs12547899	rs7841035	.0009	.0272	.2295	.7424
rs7841035	rs7826594	.0003	.2300	.4267	.3430
rs7826594	rs11785364	.4000	.0222	.0000	.5778
rs11785364	rs12550078	.0000	.4000	.0000	.6000
rs12550078	rs10282827	.0000	.0000	.0056	.9944

rs10282827	rs7815262	.0050	.0005	.2450	.7495
rs7815262	rs7818886	.0000	.2500	.0000	.7500
rs7818886	rs4909622	.0000	.0000	.6333	.3667
rs4909622	rs4243860	.4397	.1989	.0001	.3614
rs4243860	rs13270325	.0000	.4451	.0000	.5549
rs13270325	rs10454360	.0000	.0000	.3966	.6034
rs10454360	rs10112787	.0067	.3899	.4186	.1849
rs10112787	rs10454361	.0009	.4226	.0697	.5068
rs10454361	rs16906675	.0024	.0666	.0551	.8759
rs16906675	rs4481635	.0110	.0446	.3390	.6054
rs4481635	rs10088784	.3499	.0001	.1223	.5277
rs10088784	rs10505668	.3444	.1279	.0001	.5277
rs10505668	rs10093040	.0003	.3480	.1120	.5397
rs10093040	rs16906685	.0060	.1064	.0951	.7925
rs16906685	rs9324441	.0008	.1003	.4205	.4783
rs9324441	rs10505669	.0012	.4266	.0321	.5401
rs10505669	rs10096434	.0008	.0325	.3603	.6064
rs10096434	rs10095767	.0003	.3633	.1701	.4662
rs10095767	rs7827157	.0000	.1707	.0244	.8049
rs7827157	rs7827565	.0013	.0199	.3227	.6562
rs7827565	rs10093526	.0006	.3102	.1345	.5547
rs10093526	rs4314677	NA	NA	NA	NA
rs4314677	rs6651457	NA	NA	NA	NA
rs6651457	rs6651453	.0000	.0952	.0000	.9048
rs6651453	rs4382508	.0000	.0000	.4722	.5278
rs4382508	rs6986650	.0002	.4720	.0331	.4947
rs6986650	rs13281211	.0000	.0333	.0000	.9667
rs13281211	rs13273064	.0000	.0000	.0000	1.000
rs13273064	rs4243861	NA	NA	NA	NA
rs4243861	rs4243862	NA	NA	NA	NA
rs4243862	rs11986994	NA	NA	NA	NA
rs11986994	rs11994631	.0312	.0000	.0000	.9687
rs11994631	rs4517157	.0309	.0003	.2941	.6747
rs4517157	rs4645591	.0002	.3521	.1248	.5229
rs4645591	rs4507801	NA	NA	NA	NA
rs4507801	rs10096383	NA	NA	NA	NA
rs10096383	rs7844490	NA	NA	NA	NA
rs7844490	rs9650542	NA	NA	NA	NA
rs9650542	rs9650553	NA	NA	NA	NA
rs9650553	rs9650554	NA	NA	NA	NA
rs9650554	rs12676648	.0000	.4326	.0000	.5674
rs12676648	rs12543723	.0000	.0000	.1000	.9000
rs12543723	rs7845288	.0000	.1000	.0000	.9000
rs7845288	rs4460408	.0000	.0000	.0000	1.000
rs4460408	rs12545612	NA	NA	NA	NA
rs12545612	rs12542475	NA	NA	NA	NA
rs12542475	rs12547296	NA	NA	NA	NA
rs12547296	rs12676246	NA	NA	NA	NA
rs12676246	rs10087949	.0000	.1250	.0060	.8690

rs10087949	rs12545206	.0000	.0000	.1047	.8953
rs12545206	rs7820886	NA	NA	NA	NA
rs7820886	rs7843519	NA	NA	NA	NA
rs7843519	rs11784156	NA	NA	NA	NA
rs11784156	rs7821320	.0654	.0001	.5537	.3808
rs7821320	rs11780624	.0001	.6277	.1666	.2056
rs11780624	rs12155984	.0000	.1666	.6666	.1667
rs12155984	rs12156283	.5499	.1167	.0001	.3333
rs12156283	rs11984900	.1886	.3614	.0003	.4497
rs11984900	rs13267873	.1889	.0000	.0444	.7667
rs13267873	rs7014510	.0006	.2327	.2438	.5228
rs7014510	rs11989063	.0010	.2435	.1935	.5621
rs11989063	rs9644483	.0286	.1659	.0214	.7841
rs9644483	rs4295697	.0001	.0499	.4111	.5389
rs4295697	rs3903129	.4110	.0001	.1501	.4388
rs3903129	rs7819274	.0829	.4732	.0013	.4425
rs7819274	rs7834506	.0000	.0843	.0000	.9157
rs7834506	rs4366109	.0000	.0000	.4056	.5944
rs4366109	rs12544978	.0005	.4050	.0328	.5616
rs12544978	rs7010543	.0037	.0296	.0407	.9259
rs7010543	rs6991281	.0449	.0000	.0731	.8820
rs6991281	rs2076987	.1134	.0002	.2900	.5964
rs2076987	rs11781274	.0001	.4044	.0841	.5114
rs11781274	rs730453	.0003	.0830	.2330	.6836
rs730453	rs16906739	.0012	.2322	.1433	.6234
rs16906739	rs4256627	.0003	.1441	.1386	.7170
rs4256627	rs7831122	.0950	.0300	.0062	.8688
rs7831122	rs6577754	.0974	.0062	.0184	.8779
rs6577754	rs12114754	.0065	.1242	.0276	.8417
rs12114754	rs7835022	.0000	.0341	.0000	.9659
rs7835022	rs12547631	.0000	.0000	.0000	1.000
rs12547631	rs6988370	.0000	.0000	.1351	.8649
rs6988370	rs16906741	.0134	.1217	.0880	.7769
rs16906741	rs7007075	.0005	.1217	.1384	.7394
rs7007075	rs13277544	.0003	.1386	.0830	.7781
rs13277544	rs13250544	.0833	.0000	.1500	.7666
rs13250544	rs4401897	.2333	.0000	.0000	.7667
rs4401897	rs724144	.0000	.2333	.0000	.7667
rs724144	rs724145	.0000	.0000	.1111	.8889
rs724145	rs10104073	.0000	.1124	.0000	.8876
rs10104073	rs10091529	.0000	.0000	.0000	1.000
rs10091529	rs16906744	.0000	.0000	.2294	.7706
rs16906744	rs12375407	.0007	.2288	.2582	.5124
rs12375407	rs6988964	.0007	.2550	.1072	.6371
rs6988964	rs12375361	.0028	.1052	.2188	.6732
rs12375361	rs16906754	.0007	.2215	.2437	.5340
rs16906754	rs12546728	.0000	.2356	.0000	.7644
rs12546728	rs11781798	.0000	.0000	.0000	1.000
rs11781798	rs12541799	.0000	.0000	.1556	.8444

rs12541799	rs16906756	.0009	.1508	.2351	.6133
rs16906756	rs10094722	.0000	.2360	.0056	.7584
rs10094722	rs10095092	.0057	.0000	.0000	.9943
rs10095092	rs11787425	.0000	.0057	.1136	.8807
rs11787425	rs7812939	.0010	.1058	.0608	.8324
rs7812939	rs13266672	NA	NA	NA	NA
rs13266672	rs16906760	NA	NA	NA	NA
rs16906760	rs9657449	NA	NA	NA	NA
rs9657449	rs16906762	NA	NA	NA	NA
rs16906762	rs9657432	.0000	.2222	.0000	.7778
rs9657432	rs16906764	.0000	.0000	.2444	.7556
rs16906764	rs1014197	.1657	.0787	.1343	.6213
rs1014197	rs16906771	.0002	.2998	.1776	.5224
rs16906771	rs16906772	.1818	.0001	.0626	.7556
rs16906772	rs4416854	.0000	.2443	.0000	.7557
rs4416854	rs11777491	.0000	.0000	.5167	.4833
rs11777491	rs11780634	.4889	.0278	.0000	.4833
rs11780634	rs16906778	.0000	.4889	.0000	.5111
rs16906778	rs17660848	.0000	.0000	.1011	.8989
rs17660848	rs10098209	.0082	.0929	.3064	.5924
rs10098209	rs10481395	NA	NA	NA	NA
rs10481395	rs10481396	NA	NA	NA	NA
rs10481396	rs7822049	.0000	.0000	.0000	1.000
rs7822049	rs11987180	.0000	.0000	.0511	.9489
rs11987180	rs11990934	.0508	.0003	.3185	.6304
rs11990934	rs11166685	.0000	.3693	.0000	.6307
rs11166685	rs7010982	.0000	.0000	.2670	.7330
rs7010982	rs16906780	.0009	.2687	.1957	.5346
rs16906780	rs16906781	.0013	.2021	.3824	.4141
rs16906781	rs16906782	.0016	.3821	.2135	.4028
rs16906782	rs13278930	.0000	.2317	.0000	.7683
rs13278930	rs4319134	.0000	.0000	.1890	.8110
rs4319134	rs4527908	.0005	.1757	.4484	.3755
rs4527908	rs4471082	.0005	.4484	.1757	.3755
rs4471082	rs7827430	.1778	.0000	.0000	.8222
rs7827430	rs7831697	.1703	.0001	.1365	.6931
rs7831697	rs13266368	.0000	.3068	.0000	.6932
rs13266368	rs17609328	.0000	.0000	.0955	.9045

Table C.10: Haplotype frequencies gene desert, population CEU

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs4909620	rs7357354	NA	NA	NA	NA
rs7357354	rs12677926	NA	NA	NA	NA
rs12677926	rs4909621	.0000	.0000	.4345	.5655
rs4909621	rs7008788	.4167	.0000	.0000	.5833
rs7008788	rs6995856	.0001	.4127	.3720	.2152

rs6995856	rs11777213	.3875	.0000	.0000	.6125
rs11777213	rs11776500	.3841	.0000	.0000	.6159
rs11776500	rs11786769	.0001	.3850	.4137	.2012
rs11786769	rs17607388	.0000	.4157	.0169	.5674
rs17607388	rs10086749	.0000	.0167	.2056	.7778
rs10086749	rs12547899	.0048	.2007	.0285	.7659
rs12547899	rs7841035	.0000	.0333	.0000	.9667
rs7841035	rs7826594	.0000	.0000	.4167	.5833
rs7826594	rs11785364	.4167	.0000	.0000	.5833
rs11785364	rs12550078	.0000	.4167	.0000	.5833
rs12550078	rs10282827	.0000	.0000	.0333	.9667
rs10282827	rs7815262	.0000	.0333	.0000	.9667
rs7815262	rs7818886	.0000	.0000	.0000	1.000
rs7818886	rs4909622	.0000	.0000	.4167	.5833
rs4909622	rs4243860	.4345	.0000	.0000	.5655
rs4243860	rs13270325	.0004	.4387	.1216	.4394
rs13270325	rs10454360	.0005	.1260	.4092	.4643
rs10454360	rs10112787	.0004	.4054	.2113	.3828
rs10112787	rs10454361	.0002	.2141	.3634	.4223
rs10454361	rs16906675	.0000	.3636	.0000	.6364
rs16906675	rs4481635	.0000	.0000	.1944	.8056
rs4481635	rs10088784	.1944	.0000	.0167	.7889
rs10088784	rs10505668	.1944	.0167	.0000	.7889
rs10505668	rs10093040	.0004	.1941	.2052	.6004
rs10093040	rs16906685	.0001	.2055	.5832	.2112
rs16906685	rs9324441	.0001	.5832	.2055	.2112
rs9324441	rs10505669	.0008	.2047	.2047	.5897
rs10505669	rs10096434	.0037	.2018	.2074	.5870
rs10096434	rs10095767	.0000	.2111	.0000	.7889
rs10095767	rs7827157	.0000	.0000	.0000	1.000
rs7827157	rs7827565	.0000	.0000	.1646	.8354
rs7827565	rs10093526	.0004	.1663	.2175	.6158
rs10093526	rs4314677	.0004	.1950	.2123	.5923
rs4314677	rs6651457	.0000	.2111	.0000	.7889
rs6651457	rs6651453	.0000	.0000	.0000	1.000
rs6651453	rs4382508	.0000	.0000	.2111	.7889
rs4382508	rs6986650	.0037	.2074	.2018	.5870
rs6986650	rs13281211	.0020	.2002	.0205	.7773
rs13281211	rs13273064	.0000	.0225	.0000	.9775
rs13273064	rs4243861	.0000	.0000	.2056	.7944
rs4243861	rs4243862	NA	NA	NA	NA
rs4243862	rs11986994	NA	NA	NA	NA
rs11986994	rs11994631	.0000	.0000	.0000	1.000
rs11994631	rs4517157	.0000	.0000	.3833	.6167
rs4517157	rs4645591	.0625	.3208	.3708	.2458
rs4645591	rs4507801	.4233	.0002	.1708	.4057
rs4507801	rs10096383	.4046	.1906	.0002	.4046
rs10096383	rs7844490	.4101	.0000	.0000	.5899
rs7844490	rs9650542	.0001	.4100	.3482	.2417

rs9650542	rs9650553	.0002	.3482	.4830	.1687
rs9650553	rs9650554	.4775	.0000	.0000	.5225
rs9650554	rs12676648	.0003	.4772	.1570	.3655
rs12676648	rs12543723	.0000	.1556	.0556	.7889
rs12543723	rs7845288	.0000	.0511	.0227	.9261
rs7845288	rs4460408	.0227	.0000	.1421	.8352
rs4460408	rs12545612	.1680	.0005	.4500	.3815
rs12545612	rs12542475	.0000	.6180	.3820	.0000
rs12542475	rs12547296	.0000	.3807	.6193	.0000
rs12547296	rs12676246	.0117	.6077	.3577	.0230
rs12676246	rs10087949	.0035	.3687	.0021	.6257
rs10087949	rs12545206	.0056	.0000	.0000	.9944
rs12545206	rs7820886	.0000	.0056	.0278	.9667
rs7820886	rs7843519	.0278	.0000	.0000	.9722
rs7843519	rs11784156	.0000	.0241	.1566	.8193
rs11784156	rs7821320	.1566	.0000	.0301	.8133
rs7821320	rs11780624	.0001	.1999	.6166	.1834
rs11780624	rs12155984	NA	NA	NA	NA
rs12155984	rs12156283	NA	NA	NA	NA
rs12156283	rs11984900	NA	NA	NA	NA
rs11984900	rs13267873	.0278	.0000	.0000	.9722
rs13267873	rs7014510	.0000	.0278	.1222	.8500
rs7014510	rs11989063	.0004	.1218	.5607	.3171
rs11989063	rs9644483	.0003	.5608	.1664	.2725
rs9644483	rs4295697	.0082	.1585	.1251	.7082
rs4295697	rs3903129	.1332	.0001	.1834	.6832
rs3903129	rs7819274	.1721	.1446	.0002	.6832
rs7819274	rs7834506	.0000	.1722	.0222	.8056
rs7834506	rs4366109	.0051	.0171	.0227	.9551
rs4366109	rs12544978	.0000	.0278	.0056	.9667
rs12544978	rs7010543	.0000	.0056	.0000	.9944
rs7010543	rs6991281	.0000	.0000	.0000	1.000
rs6991281	rs2076987	.0000	.0000	.0281	.9719
rs2076987	rs11781274	.0000	.0281	.1404	.8315
rs11781274	rs730453	.0044	.1361	.1192	.7403
rs730453	rs16906739	.0000	.1222	.0111	.8667
rs16906739	rs4256627	.0004	.0107	.5607	.4282
rs4256627	rs7831122	.5640	.0000	.0000	.4360
rs7831122	rs6577754	.5783	.0000	.0000	.4217
rs6577754	rs12114754	.0000	.5747	.0000	.4253
rs12114754	rs7835022	.0000	.0000	.0000	1.000
rs7835022	rs12547631	.0000	.0000	.0056	.9944
rs12547631	rs6988370	.0000	.0060	.0000	.9940
rs6988370	rs16906741	.0000	.0000	.0119	.9881
rs16906741	rs7007075	.0004	.0107	.5607	.4282
rs7007075	rs13277544	.0011	.5607	.0270	.4112
rs13277544	rs13250544	.0281	.0000	.0000	.9719
rs13250544	rs4401897	.0278	.0000	.0000	.9722
rs4401897	rs724144	.0000	.0298	.0000	.9702

rs724144	rs724145	.0000	.0000	.5595	.4405
rs724145	rs10104073	.0000	.5611	.0000	.4389
rs10104073	rs10091529	.0000	.0000	.0000	1.000
rs10091529	rs16906744	.0000	.0000	.1193	.8807
rs16906744	rs12375407	.0026	.1168	.0259	.8548
rs12375407	rs6988964	.0014	.0319	.5708	.3958
rs6988964	rs12375361	.0014	.5708	.0319	.3958
rs12375361	rs16906754	.0000	.0333	.1278	.8389
rs16906754	rs12546728	.0000	.1278	.0000	.8722
rs12546728	rs11781798	.0000	.0000	.0000	1.000
rs11781798	rs12541799	.0000	.0000	.0167	.9833
rs12541799	rs16906756	.0000	.0167	.0889	.8944
rs16906756	rs10094722	.0000	.0889	.0000	.9111
rs10094722	rs10095092	.0000	.0000	.0000	1.000
rs10095092	rs11787425	.0000	.0000	.6517	.3483
rs11787425	rs7812939	.0000	.6444	.0000	.3556
rs7812939	rs13266672	.0000	.0000	.0333	.9667
rs13266672	rs16906760	.0333	.0000	.1056	.8611
rs16906760	rs9657449	.0000	.1348	.0000	.8652
rs9657449	rs16906762	.0000	.0000	.0281	.9719
rs16906762	rs9657432	.0000	.0333	.0000	.9667
rs9657432	rs16906764	.0000	.0000	.0333	.9667
rs16906764	rs1014197	.0000	.0333	.1056	.8611
rs1014197	rs16906771	.0000	.1056	.0167	.8778
rs16906771	rs16906772	.0166	.0001	.1223	.8610
rs16906772	rs4416854	.0000	.1389	.0000	.8611
rs4416854	rs11777491	.0000	.0000	.2722	.7278
rs11777491	rs11780634	.2722	.0000	.0000	.7278
rs11780634	rs16906778	.0000	.2722	.0000	.7278
rs16906778	rs17660848	.0000	.0000	.6193	.3807
rs17660848	rs10098209	.0001	.6192	.1476	.2331
rs10098209	rs10481395	.0000	.1264	.0000	.8736
rs10481395	rs10481396	.0000	.0000	.0000	1.000
rs10481396	rs7822049	.0000	.0000	.1444	.8556
rs7822049	rs11987180	.0037	.1407	.0074	.8481
rs11987180	rs11990934	.0108	.0003	.2281	.7608
rs11990934	rs11166685	.0044	.2372	.1136	.6449
rs11166685	rs7010982	.0102	.1078	.1359	.7461
rs7010982	rs16906780	.0000	.1529	.7352	.1118
rs16906780	rs16906781	.0001	.7352	.1175	.1472
rs16906781	rs16906782	.0000	.1111	.0944	.7944
rs16906782	rs13278930	.0038	.0951	.0137	.8875
rs13278930	rs4319134	.0002	.0174	.7351	.2473
rs4319134	rs4527908	.0070	.7178	.1391	.1362
rs4527908	rs4471082	.0002	.1459	.7189	.1350
rs4471082	rs7827430	.7167	.0000	.0000	.2833
rs7827430	rs7831697	.7202	.0000	.0179	.2619
rs7831697	rs13266368	.0000	.7439	.0000	.2561
rs13266368	rs17609328	.0000	.0000	.6307	.3693

Table C.11: Haplotype frequencies gene desert, population CHB

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs4909620	rs7357354	.1538	.0257	.0000	.8205
rs7357354	rs12677926	.0000	.1538	.0000	.8462
rs12677926	rs4909621	.0000	.0000	.2222	.7778
rs4909621	rs7008788	.2222	.0000	.0000	.7778
rs7008788	rs6995856	.0008	.2135	.4159	.3699
rs6995856	rs11777213	.4000	.0125	.0000	.5875
rs11777213	rs11776500	.4103	.0000	.0128	.5769
rs11776500	rs11786769	.0005	.4263	.2068	.3664
rs11786769	rs17607388	.0000	.2222	.0000	.7778
rs17607388	rs10086749	.0000	.0000	.3444	.6556
rs10086749	rs12547899	.0001	.3443	.0665	.5890
rs12547899	rs7841035	.0000	.0667	.0000	.9333
rs7841035	rs7826594	.0000	.0000	.2222	.7778
rs7826594	rs11785364	.2222	.0000	.0000	.7778
rs11785364	rs12550078	.0000	.2222	.0000	.7778
rs12550078	rs10282827	.0000	.0000	.0455	.9545
rs10282827	rs7815262	.0000	.0455	.0000	.9545
rs7815262	rs7818886	.0000	.0000	.0000	1.000
rs7818886	rs4909622	.0000	.0000	.2222	.7778
rs4909622	rs4243860	.2564	.0000	.0000	.7436
rs4243860	rs13270325	.0064	.2436	.1741	.5759
rs13270325	rs10454360	.0052	.1853	.2091	.6004
rs10454360	rs10112787	.0002	.2220	.3665	.4113
rs10112787	rs10454361	.0001	.4165	.4026	.1807
rs10454361	rs16906675	.0000	.4028	.0000	.5972
rs16906675	rs4481635	.0000	.0000	.1222	.8778
rs4481635	rs10088784	.1222	.0000	.0000	.8778
rs10088784	rs10505668	.1222	.0000	.0000	.8778
rs10505668	rs10093040	.0008	.1215	.3659	.5119
rs10093040	rs16906685	.0000	.3666	.5111	.1223
rs16906685	rs9324441	.0000	.5111	.3666	.1223
rs9324441	rs10505669	.0002	.3665	.0887	.5446
rs10505669	rs10096434	.0000	.0889	.1222	.7889
rs10096434	rs10095767	.0000	.1222	.0000	.8778
rs10095767	rs7827157	.0000	.0000	.0000	1.000
rs7827157	rs7827565	.0000	.0000	.1125	.8875
rs7827565	rs10093526	.0005	.1149	.3456	.5390
rs10093526	rs4314677	.0008	.3628	.1242	.5122
rs4314677	rs6651457	.0000	.1250	.0000	.8750
rs6651457	rs6651453	.0000	.0000	.0000	1.000
rs6651453	rs4382508	.0000	.0000	.1222	.8778
rs4382508	rs6986650	.0000	.1222	.0889	.7889

rs6986650	rs13281211	.0000	.0889	.0000	.9111
rs13281211	rs13273064	.0000	.0000	.0000	1.000
rs13273064	rs4243861	.0000	.0000	.1222	.8778
rs4243861	rs4243862	.1222	.0000	.0000	.8778
rs4243862	rs11986994	.0000	.1222	.0000	.8778
rs11986994	rs11994631	.0000	.0000	.0000	1.000
rs11994631	rs4517157	.0000	.0000	.4778	.5222
rs4517157	rs4645591	.0001	.4777	.2999	.2223
rs4645591	rs4507801	.2954	.0000	.0682	.6364
rs4507801	rs10096383	.3295	.0341	.0000	.6363
rs10096383	rs7844490	.3295	.0000	.0000	.6705
rs7844490	rs9650542	.0001	.3295	.5454	.1251
rs9650542	rs9650553	.0000	.5333	.4444	.0222
rs9650553	rs9650554	.4444	.0000	.0000	.5556
rs9650554	rs12676648	.0000	.4444	.4778	.0778
rs12676648	rs12543723	.0004	.4774	.1107	.4115
rs12543723	rs7845288	.0000	.1111	.0000	.8889
rs7845288	rs4460408	.0000	.0000	.0000	1.000
rs4460408	rs12545612	.0000	.0000	.3667	.6333
rs12545612	rs12542475	.0000	.3667	.6333	.0000
rs12542475	rs12547296	.0000	.6333	.3667	.0000
rs12547296	rs12676246	.0112	.3554	.6221	.0112
rs12676246	rs10087949	.0000	.6333	.0000	.3667
rs10087949	rs12545206	.0000	.0000	.0000	1.000
rs12545206	rs7820886	.0000	.0000	.0556	.9444
rs7820886	rs7843519	.0455	.0000	.0000	.9545
rs7843519	rs11784156	.0105	.0252	.0490	.9153
rs11784156	rs7821320	.0581	.0000	.0465	.8953
rs7821320	rs11780624	.0008	.1103	.3659	.5230
rs11780624	rs12155984	.0008	.3659	.1103	.5230
rs12155984	rs12156283	.1111	.0000	.0000	.8889
rs12156283	rs11984900	.0568	.0568	.0000	.8864
rs11984900	rs13267873	.0568	.0000	.0000	.9432
rs13267873	rs7014510	.0002	.0553	.5664	.3780
rs7014510	rs11989063	.0000	.5666	.3333	.1000
rs11989063	rs9644483	.0052	.3282	.0393	.6274
rs9644483	rs4295697	.0090	.0355	.0466	.9090
rs4295697	rs3903129	.0556	.0000	.0444	.9000
rs3903129	rs7819274	.0444	.0556	.0000	.9000
rs7819274	rs7834506	.0000	.0444	.0000	.9556
rs7834506	rs4366109	.0000	.0000	.0556	.9444
rs4366109	rs12544978	.0000	.0556	.0000	.9444
rs12544978	rs7010543	.0000	.0000	.0000	1.000
rs7010543	rs6991281	.0000	.0000	.0000	1.000
rs6991281	rs2076987	.0000	.0000	.0556	.9444
rs2076987	rs11781274	.0090	.0466	.0355	.9090
rs11781274	rs730453	.0002	.0443	.5665	.3891
rs730453	rs16906739	.0000	.5667	.0000	.4333
rs16906739	rs4256627	.0000	.0000	.3333	.6667

rs4256627	rs7831122	.3295	.0000	.0114	.6591
rs7831122	rs6577754	.3295	.0114	.0000	.6591
rs6577754	rs12114754	.0000	.3295	.0000	.6705
rs12114754	rs7835022	.0000	.0000	.0000	1.000
rs7835022	rs12547631	.0000	.0000	.0000	1.000
rs12547631	rs6988370	.0000	.0000	.0000	1.000
rs6988370	rs16906741	.0000	.0000	.0000	1.000
rs16906741	rs7007075	.0000	.0000	.3333	.6667
rs7007075	rs13277544	.0005	.3290	.0449	.6255
rs13277544	rs13250544	.0455	.0000	.0000	.9545
rs13250544	rs4401897	.0556	.0000	.0000	.9444
rs4401897	rs724144	.0000	.0625	.0000	.9375
rs724144	rs724145	.0000	.0000	.3250	.6750
rs724145	rs10104073	.0000	.3333	.0000	.6667
rs10104073	rs10091529	.0000	.0000	.0000	1.000
rs10091529	rs16906744	.0000	.0000	.5568	.4432
rs16906744	rs12375407	.0003	.5565	.0565	.3866
rs12375407	rs6988964	.0015	.0541	.3652	.5793
rs6988964	rs12375361	.0015	.3652	.0541	.5793
rs12375361	rs16906754	.0209	.0347	.0347	.9097
rs16906754	rs12546728	.0000	.0568	.0000	.9432
rs12546728	rs11781798	.0000	.0000	.0000	1.000
rs11781798	rs12541799	.0000	.0000	.0000	1.000
rs12541799	rs16906756	.0000	.0000	.5111	.4889
rs16906756	rs10094722	.0000	.5111	.0000	.4889
rs10094722	rs10095092	.0000	.0000	.0000	1.000
rs10095092	rs11787425	.0000	.0000	.3778	.6222
rs11787425	rs7812939	.0000	.3778	.0000	.6222
rs7812939	rs13266672	.0000	.0000	.0556	.9444
rs13266672	rs16906760	.0556	.0000	.0000	.9444
rs16906760	rs9657449	.0000	.0238	.0000	.9762
rs9657449	rs16906762	.0000	.0000	.0238	.9762
rs16906762	rs9657432	.0000	.0556	.0000	.9444
rs9657432	rs16906764	.0000	.0000	.0568	.9432
rs16906764	rs1014197	.0000	.0568	.5114	.4318
rs1014197	rs16906771	.0000	.5000	.0000	.5000
rs16906771	rs16906772	.0000	.0000	.0889	.9111
rs16906772	rs4416854	.0000	.0889	.0000	.9111
rs4416854	rs11777491	.0000	.0000	.1591	.8409
rs11777491	rs11780634	.1591	.0000	.0000	.8409
rs11780634	rs16906778	.1000	.0556	.0000	.8444
rs16906778	rs17660848	.0006	.0994	.3883	.5117
rs17660848	rs10098209	.0006	.3883	.0994	.5117
rs10098209	rs10481395	.0000	.1000	.0000	.9000
rs10481395	rs10481396	.0000	.0000	.0000	1.000
rs10481396	rs7822049	.0000	.0000	.0000	1.000
rs7822049	rs11987180	.0000	.0000	.0000	1.000
rs11987180	rs11990934	.0000	.0000	.1023	.8977
rs11990934	rs11166685	.0000	.1023	.0000	.8977

rs11166685	rs7010982	.0000	.0000	.1000	.9000
rs7010982	rs16906780	.0006	.1041	.3948	.5006
rs16906780	rs16906781	.0000	.3953	.4767	.1279
rs16906781	rs16906782	.0000	.4556	.0000	.5444
rs16906782	rs13278930	.0000	.0000	.0125	.9875
rs13278930	rs4319134	.0117	.0011	.3472	.6400
rs4319134	rs4527908	.0000	.3636	.5227	.1137
rs4527908	rs4471082	.0122	.5100	.3767	.1011
rs4471082	rs7827430	.3889	.0000	.0000	.6111
rs7827430	rs7831697	.3571	.0238	.0000	.6190
rs7831697	rs13266368	.0000	.3659	.0000	.6341
rs13266368	rs17609328	.0000	.0000	.3523	.6477

Table C.12: Haplotype frequencies gene desert, population JPT

SNP 1	SNP 2	η_0	η_1	η_2	η_3
rs4909620	rs7357354	.1667	.0833	.0000	.7500
rs7357354	rs12677926	.0000	.1625	.0000	.8375
rs12677926	rs4909621	.0000	.0000	.2561	.7439
rs4909621	rs7008788	.2614	.0000	.0000	.7386
rs7008788	rs6995856	.0000	.2558	.3139	.4303
rs6995856	rs11777213	.3293	.0000	.0000	.6707
rs11777213	rs11776500	.3256	.0116	.0000	.6628
rs11776500	rs11786769	.0003	.3108	.2664	.4225
rs11786769	rs17607388	.0000	.2667	.0000	.7333
rs17607388	rs10086749	.0000	.0000	.4091	.5909
rs10086749	rs12547899	.0013	.4078	.0896	.5013
rs12547899	rs7841035	.0000	.0889	.0000	.9111
rs7841035	rs7826594	.0000	.0000	.2614	.7386
rs7826594	rs11785364	.2614	.0000	.0000	.7386
rs11785364	rs12550078	.0000	.2667	.0000	.7333
rs12550078	rs10282827	.0000	.0000	.0909	.9091
rs10282827	rs7815262	.0000	.0909	.0000	.9091
rs7815262	rs7818886	.0000	.0000	.0000	1.000
rs7818886	rs4909622	.0000	.0000	.2791	.7209
rs4909622	rs4243860	.3000	.0000	.0000	.7000
rs4243860	rs13270325	.0007	.2993	.0993	.6007
rs13270325	rs10454360	.0006	.1041	.2785	.6169
rs10454360	rs10112787	.0012	.2895	.3825	.3268
rs10112787	rs10454361	.0002	.3998	.3855	.2145
rs10454361	rs16906675	.0000	.3649	.0000	.6351
rs16906675	rs4481635	.0000	.0000	.1136	.8864
rs4481635	rs10088784	.1136	.0000	.0000	.8864
rs10088784	rs10505668	.1136	.0000	.0000	.8864
rs10505668	rs10093040	.0046	.1091	.4045	.4819
rs10093040	rs16906685	.0001	.4090	.4885	.1024
rs16906685	rs9324441	.0001	.4885	.4090	.1024

rs9324441	rs10505669	.0018	.4073	.1119	.4790
rs10505669	rs10096434	.0088	.1048	.1162	.7702
rs10096434	rs10095767	.0000	.1250	.0000	.8750
rs10095767	rs7827157	.0000	.0000	.0000	1.000
rs7827157	rs7827565	.0000	.0000	.0930	.9070
rs7827565	rs10093526	.0030	.0900	.4039	.5030
rs10093526	rs4314677	.0046	.4045	.1091	.4819
rs4314677	rs6651457	.0000	.1136	.0000	.8864
rs6651457	rs6651453	.0000	.0000	.0000	1.000
rs6651453	rs4382508	.0000	.0000	.1111	.8889
rs4382508	rs6986650	.0088	.1023	.1134	.7755
rs6986650	rs13281211	.0000	.1222	.0000	.8778
rs13281211	rs13273064	.0000	.0000	.0000	1.000
rs13273064	rs4243861	.0000	.0000	.1136	.8864
rs4243861	rs4243862	.1136	.0000	.0000	.8864
rs4243862	rs11986994	.0000	.1136	.0000	.8864
rs11986994	rs11994631	.0000	.0000	.0000	1.000
rs11994631	rs4517157	.0000	.0000	.6023	.3977
rs4517157	rs4645591	.0001	.6045	.1976	.1978
rs4645591	rs4507801	.1854	.0123	.0471	.7552
rs4507801	rs10096383	.1818	.0455	.0000	.7727
rs10096383	rs7844490	.1744	.0000	.0000	.8256
rs7844490	rs9650542	.0002	.1743	.6394	.1862
rs9650542	rs9650553	.0001	.6363	.3181	.0455
rs9650553	rs9650554	.3182	.0000	.0000	.6818
rs9650554	rs12676648	.0001	.3181	.5794	.1024
rs12676648	rs12543723	.0003	.5792	.1361	.2844
rs12543723	rs7845288	.0000	.1364	.0000	.8636
rs7845288	rs4460408	.0000	.0000	.0000	1.000
rs4460408	rs12545612	.0000	.0000	.2791	.7209
rs12545612	rs12542475	.0000	.2841	.7159	.0000
rs12542475	rs12547296	.0000	.7317	.2683	.0000
rs12547296	rs12676246	.0000	.2683	.7317	.0000
rs12676246	rs10087949	.0000	.7125	.0000	.2875
rs10087949	rs12545206	.0000	.0000	.0000	1.000
rs12545206	rs7820886	.0000	.0000	.0395	.9605
rs7820886	rs7843519	.0385	.0000	.0000	.9615
rs7843519	rs11784156	.0000	.0789	.0526	.8684
rs11784156	rs7821320	.0540	.0000	.0676	.8783
rs7821320	rs11780624	.0007	.1129	.3516	.5348
rs11780624	rs12155984	.0007	.3516	.1129	.5348
rs12155984	rs12156283	.1136	.0000	.0000	.8864
rs12156283	rs11984900	.0568	.0568	.0000	.8863
rs11984900	rs13267873	.0568	.0000	.0000	.9432
rs13267873	rs7014510	.0009	.0573	.5456	.3962
rs7014510	rs11989063	.0001	.5464	.3836	.0698
rs11989063	rs9644483	.0000	.3864	.0114	.6023
rs9644483	rs4295697	.0000	.0114	.0568	.9318
rs4295697	rs3903129	.0568	.0000	.0114	.9318

rs3903129	rs7819274	.0113	.0569	.0000	.9318
rs7819274	rs7834506	.0000	.0114	.0000	.9886
rs7834506	rs4366109	.0000	.0000	.0568	.9432
rs4366109	rs12544978	.0000	.0568	.0000	.9432
rs12544978	rs7010543	.0000	.0000	.0000	1.000
rs7010543	rs6991281	.0000	.0000	.0000	1.000
rs6991281	rs2076987	.0000	.0000	.0667	.9333
rs2076987	rs11781274	.0000	.0667	.0111	.9222
rs11781274	rs730453	.0052	.0062	.5403	.4483
rs730453	rs16906739	.0000	.5455	.0000	.4545
rs16906739	rs4256627	.0000	.0000	.3864	.6136
rs4256627	rs7831122	.3810	.0000	.0000	.6190
rs7831122	rs6577754	.3810	.0000	.0000	.6190
rs6577754	rs12114754	.0000	.3864	.0000	.6136
rs12114754	rs7835022	.0000	.0000	.0000	1.000
rs7835022	rs12547631	.0000	.0000	.0000	1.000
rs12547631	rs6988370	.0000	.0000	.0000	1.000
rs6988370	rs16906741	.0000	.0000	.0000	1.000
rs16906741	rs7007075	.0000	.0000	.3864	.6136
rs7007075	rs13277544	.0001	.3862	.0567	.5570
rs13277544	rs13250544	.0698	.0000	.0000	.9302
rs13250544	rs4401897	.0595	.0000	.0000	.9405
rs4401897	rs724144	.0000	.0488	.0000	.9512
rs724144	rs724145	.0000	.0000	.4024	.5976
rs724145	rs10104073	.0000	.3864	.0000	.6136
rs10104073	rs10091529	.0000	.0000	.0000	1.000
rs10091529	rs16906744	.0000	.0000	.5455	.4545
rs16906744	rs12375407	.0014	.5441	.0668	.3878
rs12375407	rs6988964	.0004	.0662	.3996	.5338
rs6988964	rs12375361	.0013	.4077	.0441	.5468
rs12375361	rs16906754	.0000	.0455	.0568	.8977
rs16906754	rs12546728	.0000	.0556	.0000	.9444
rs12546728	rs11781798	.0000	.0000	.0000	1.000
rs11781798	rs12541799	.0000	.0000	.0000	1.000
rs12541799	rs16906756	.0000	.0000	.4889	.5111
rs16906756	rs10094722	.0000	.4889	.0000	.5111
rs10094722	rs10095092	.0000	.0000	.0000	1.000
rs10095092	rs11787425	.0000	.0000	.4318	.5682
rs11787425	rs7812939	.0000	.4318	.0000	.5682
rs7812939	rs13266672	.0000	.0000	.0556	.9444
rs13266672	rs16906760	.0556	.0000	.0000	.9444
rs16906760	rs9657449	.0000	.0455	.0000	.9545
rs9657449	rs16906762	.0000	.0000	.0227	.9773
rs16906762	rs9657432	.0000	.0333	.0000	.9667
rs9657432	rs16906764	.0000	.0000	.0333	.9667
rs16906764	rs1014197	.0000	.0227	.4545	.5227
rs1014197	rs16906771	.0000	.4545	.0000	.5455
rs16906771	rs16906772	.0000	.0000	.1556	.8444
rs16906772	rs4416854	.0000	.1556	.0000	.8444

rs4416854	rs11777491	.0000	.0000	.2222	.7778
rs11777491	rs11780634	.1818	.0341	.0000	.7841
rs11780634	rs16906778	.1591	.0227	.0000	.8182
rs16906778	rs17660848	.0003	.1552	.3774	.4670
rs17660848	rs10098209	.0003	.3861	.1588	.4548
rs10098209	rs10481395	.0000	.1591	.0000	.8409
rs10481395	rs10481396	.0000	.0000	.0000	1.000
rs10481396	rs7822049	.0000	.0000	.0000	1.000
rs7822049	rs11987180	.0000	.0000	.0000	1.000
rs11987180	rs11990934	.0000	.0000	.1628	.8372
rs11990934	rs11166685	.0000	.1591	.0000	.8409
rs11166685	rs7010982	.0000	.0000	.1591	.8409
rs7010982	rs16906780	.0003	.1747	.3747	.4503
rs16906780	rs16906781	.0001	.3657	.4389	.1952
rs16906781	rs16906782	.0000	.4000	.0000	.6000
rs16906782	rs13278930	.0000	.0000	.0000	1.000
rs13278930	rs4319134	.0000	.0000	.3714	.6286
rs4319134	rs4527908	.0001	.3624	.4874	.1501
rs4527908	rs4471082	.0150	.4501	.3687	.1661
rs4471082	rs7827430	.3864	.0000	.0000	.6136
rs7827430	rs7831697	.3816	.0000	.0000	.6184
rs7831697	rs13266368	.0000	.3718	.0000	.6282
rs13266368	rs17609328	.0000	.0000	.3667	.6333