Phasing genotype data

Acknowledgement: Several slides based on a lecture course given by
Jonathan Marchini & Chris Spencer, Cape Town 2007

Further concepts

Hardy-Weinberg Principle: Alleles of an individual are chosen randomly and independently from the alleles present at the previous generation (randomly mating population)

When the Hardy-Weinberg Principle applies, the expected allele frequencies are the same from one generation to the next (Hardy-Weinberg Equilibrium).

The proportion of each genotype (AA, Aa, aa) is p^2 $2pq$ and q^2

where p and q are the frequencies of A and a, respectively

The haplotyping problem: given a set of genotypes, infer the set of haplotypes that produced the genotype

Motivation

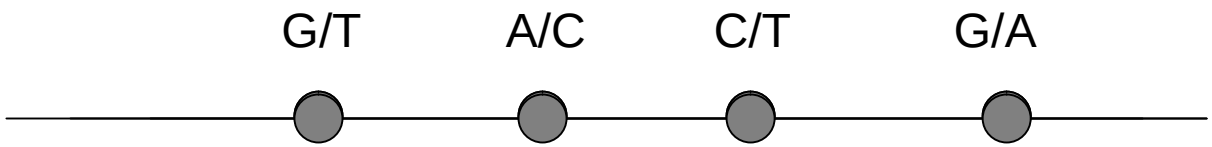
existence of haplotypes reduces the number of SNPs for genotyping

haplotypes are useful for disease association but genotyping is far cheaper than determining haplotypes experimentally

haplotyping programs have high accuracy under specific conditions – may not be worth the extra expense of experimental haplotyping

Linkage, haplotypes and HapMap

Haplotype: A set of alleles co-occurring on a chromosome (contraction of haploid genotype)



G A T G

T C T G

Clark's Algorithm (1990)

Parsimony-like approach

Start with the trivially resolvable haplotypes (by taking individuals who are heterozygous for at most one SNP)

if there are no such individuals the algorithm cannot start

For each unresolved haplotype consider whether it can be considered to be a combination of one of the trivially resolvable haplotypes and its complement

Continue until all are resolved or no further resolution possible

NB: Not guaranteed to find a solution

Solution could underestimate the true number of haplotypes

Worked example

Suppose we have genotype data for a set of SNPs

Encode the data as follows:

XX -> 0

YY -> 1

XY -> 2

Where X and Y are alleles of the SNPs (arbitrarily assigned)

Worked example

Use Clarke's algorithm to infer a parsimonious set of haplotypes that could give rise to these genotypes

The alleles of a biallelic SNP are encoded (arbitrarily) as 0, 1;
Genotypes encoded as: 00 => 0; 01 => 1; 11 => 2

SNPs:	1	2	3	4	5	6	7
	1	1	1	0	1	0	1
	1	1	0	1	1	0	1
	1	2	1	0	0	0	0
	2	2	2	0	0	1	0
	1	2	0	1	0	0	0

Maximum likelihood

Given a set of genotype data, $g_1..g_n$ from n individuals. Consider the (unknown) set of haplotypes, $h_1 \ldots h_n$ corresponding to these genotypes, occurring at frequencies $\theta_1 \ldots \theta_n$.

Maximum likelihood methods estimate $\theta_1 \ldots \theta_n$ by maximizing

$$P(g_1 \ldots g_n \mid \theta_1 \ldots \theta_n) = \prod_{i=1}^{n} P(g_i \mid \theta_1 \ldots \theta_n) = \prod_{i=1}^{n} \sum_{(h_1, h_2) \in H_i} \theta_{h_1} \theta_{h_2}$$

where H_i represents the set of (ordered) haplotype pairs consistent with genotype i

Example

Consider a set of 3 locus genotypes for which we wish to infer the underlying haplotypes

g_1	1 1 2
g_2	1 1 1
g_3	2 0 1
g_4	0 2 1

Example

First we list all possible haplotypes consistent with the genotypes. Then with each haplotype we associate a parameter which is the population frequency of the haplotype in the population.

Haplotypes		Population Frequencies (Parameters)
h_1	0 0 0	θ_1
h_2	0 0 1	θ_2
h_3	0 1 0	θ_3
h_4	0 1 1	θ_4
h_5	1 0 0	θ_5
h_6	1 0 1	θ_6
h_7	1 1 0	θ_7
h_8	1 1 1	θ_8

The Model

Consider the first genotype g_1 1 1 2

It is consistent with the pairs of haplotypes (h_2,h_8), (h_8,h_2), (h_4,h_6) (h_6,h_4)

Under the Assumption of Hardy-Weinberg equilibrium the probability of the genotype is

$$P(g_1 \mid \theta) = 2\theta_2\theta_8 + 2\theta_4\theta_6$$

The probability of all the genotypes together is the product of the probabilities of each individual genotype i.e.

$$L(\theta) = P(\text{Data} \mid \theta) = P(g_1 \mid \theta)P(g_2 \mid \theta).....P(g_n \mid \theta)$$

Maximum likelihood implementation: the EM algorithm

Implementations of maximum likelihood estimation for haplotype frequencies, and haplotype reconstructions typically use the EM algorithm.

EM initialization

Initialize EM algorithm, at some starting point consisting of a set of population haplotype frequencies. Starting point can be random or informed by our intuition for what the answer "ought" to look like.

Observed data	Missing data
g_1	$h_1 \rightarrow \theta_1$
g_2	$h_2 \rightarrow \theta_2$
⋮	⋮
g_N	$h_M \rightarrow \theta_M$

Initialization:

$$\theta_1 = q_1$$
$$\theta_2 = q_2$$
$$\vdots$$
$$\theta_M = q_M$$

$$(\Sigma_M\, q_j = 1)$$

E(xpectation) step

Assuming the current population frequencies are correct, calculate the *expected* number of copies of each haplotype in each individual under the model:

$Q^{(t)}$

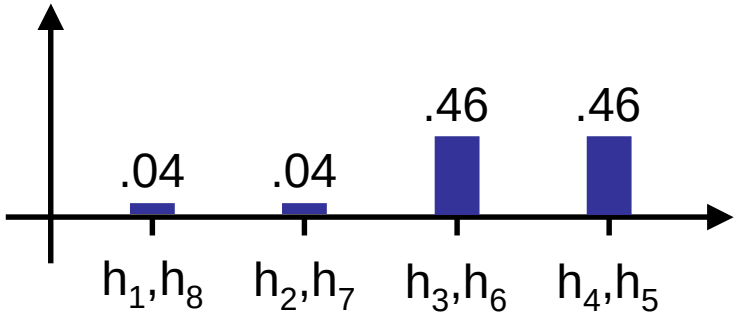
$q_1^{(t)} = q_7^{(t)} = .031$

$q_2^{(t)} = q_8^{(t)} = .094$

$q_3^{(t)} = q_5^{(t)} = .156$

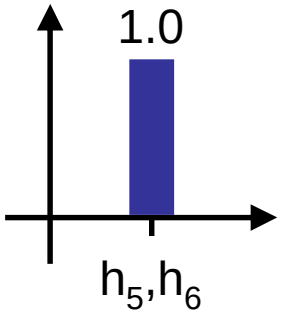
$q_4^{(t)} = q_6^{(t)} = .219$

$P(h_a, h_b \mid g_2, Q^{(t)})$



$P(h_a, h_b \mid g_3, Q^{(t)})$

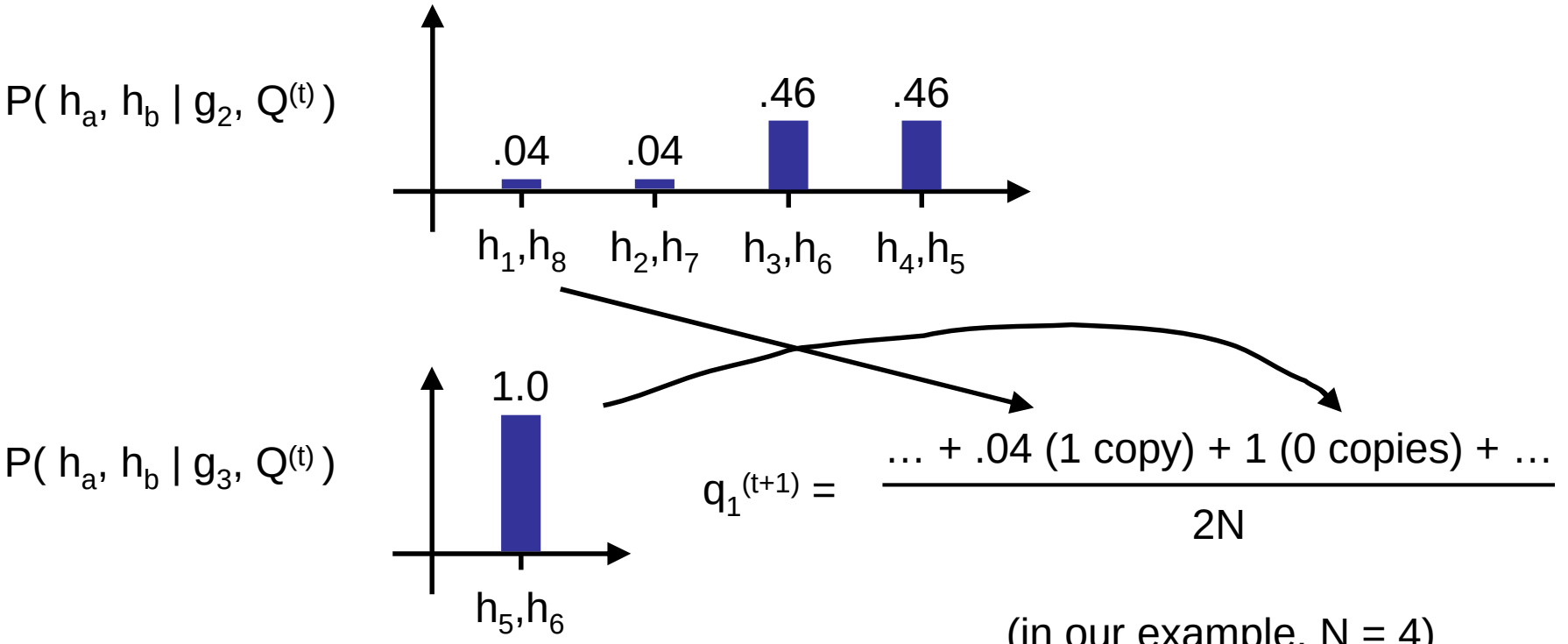
unambiguous individual



Conditioning on genotype (g_i) tells us to assign each incompatible haplotype pair a probability of zero, and to normalize such that each individual's distribution sums to one.

M(aximization) step

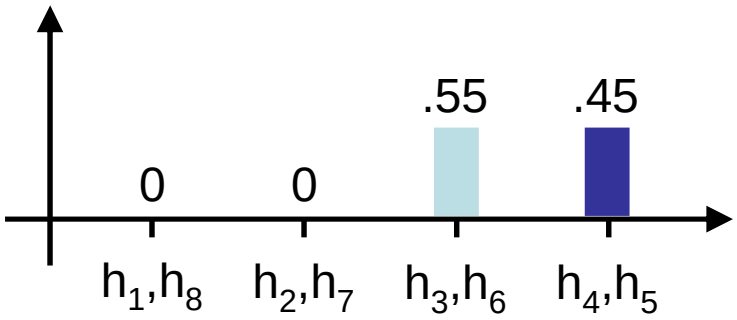
Re-estimate the parameter vector Q using our current (probabilistic) guesses about the haplotypes carried by the individuals in our sample. We compute each haplotype's new frequency $q_j^{(t+1)}$ by taking weighted counts across the $P(h_a, h_b \mid g_i, Q^{(t)})$ distributions for all individuals:



Choosing the “best” haplotype pairs

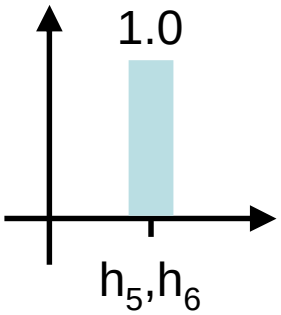
After the E and M steps have been repeated T times, the algorithm will converge on a set of haplotype frequencies. We then use these frequencies to choose a haplotype pair for each individual by identifying the most likely configuration under the model:

$P(h_a, h_b \mid g_2, Q^{(T)})$



= inferred haplotype pair

$P(h_a, h_b \mid g_3, Q^{(T)})$



Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556

Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556
1	.031	.094	.156	.219	.156	.219	.031	.094	-12.224

Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556
1	.031	.094	.156	.219	.156	.219	.031	.094	-12.224
2	.005	.024	.183	.288	.183	.288	.005	.024	-10.620

Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556
1	.031	.094	.156	.219	.156	.219	.031	.094	-12.224
2	.005	.024	.183	.288	.183	.288	.005	.024	-10.620
3	.000	.001	.187	.311	.187	.311	.000	.001	-10.163

Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556
1	.031	.094	.156	.219	.156	.219	.031	.094	-12.224
2	.005	.024	.183	.288	.183	.288	.005	.024	-10.620
3	.000	.001	.187	.311	.187	.311	.000	.001	-10.163
4	.000	.000	.187	.312	.187	.312	.000	.000	-10.145

Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556
1	.031	.094	.156	.219	.156	.219	.031	.094	-12.224
2	.005	.024	.183	.288	.183	.288	.005	.024	-10.620
3	.000	.001	.187	.311	.187	.311	.000	.001	-10.163
4	.000	.000	.187	.312	.187	.312	.000	.000	-10.145
5	.000	.000	.187	.312	.187	.312	.000	.000	-10.145

Example: EM algorithm applied to the four genotypes seen earlier

Iteration	000 θ_1	001 θ_2	010 θ_3	011 θ_4	100 θ_5	101 θ_6	110 θ_7	111 θ_8	log(L)
0	.125	.125	.125	.125	.125	.125	.125	.125	-14.556
1	.031	.094	.156	.219	.156	.219	.031	.094	-12.224
2	.005	.024	.183	.288	.183	.288	.005	.024	-10.620
3	.000	.001	.187	.311	.187	.311	.000	.001	-10.163
4	.000	.000	.187	.312	.187	.312	.000	.000	-10.145
5	.000	.000	.187	.312	.187	.312	.000	.000	-10.145

Point estimates:

g_1 $\frac{0\,1\,1}{1\,0\,1}$ g_2 $\frac{0\,1\,0}{1\,0\,1}$ g_3 $\frac{1\,0\,0}{1\,0\,1}$ g_4 $\frac{0\,1\,0}{0\,1\,1}$

This ML method breaks down badly for large numbers of SNPs, because the number of possible haplotypes increases rapidly

A Hidden Markov Model approach: Scheet and Stephens (AJHG 2006)

Cluster model for haplotypes:

Given n haplotypes $(h_1, \ldots, h_n)$ with data at M markers such that the allele at marker m of haplotype i is h_{im}

Arbitrarily label alleles 1,0. Assume that there are K, (known) haplotype clusters with distinct allele frequencies at each of the markers so that

$$P(h_i|z_i, \theta) = \prod_{m=1}^{M} \theta_{km}^{h_{im}} (1 - \theta_{km})^{1-h_{im}}$$

where z_i indicates the cluster membership of haplotype i and θ_{km} is the frequency of allele 1 of marker m in cluster k

For unknown cluster membership:

$$
\begin{aligned}
P(h_i|\alpha,\theta) &= \sum_{k=1}^{K} P(z_i = k|\alpha) P(h_i|z_i = k,\theta) \\
&= \sum_{k=1}^{K} \alpha_{\mathbf{k}} \prod_{m=1}^{M} \theta_{km}^{h_{im}} (1-\theta_{km})^{1-h_{im}}
\end{aligned}
$$

where α is the vector of cluster frequencies

This model makes the assumption of independence across markers, given the cluster. In practise marker frequencies within a cluster (i.e. the vector θ_k) tends to consist mostly of values close to 0 and 1.

The idea is that the clusters capture the variation of alleles which is not independent across markers. The remaining variation within a cluster is independent across markers (e.g. this includes a component due to genotyping error)

Introducing recombination into the model

Allow alleles, rather than haplotypes, to be sampled from clusters. Cluster membership can change along the chromosome.

Cluster transitions described by a Markov chain

$$
\begin{aligned}
P(z_{i1} = k) &= \alpha_{k1} \\
P_m(k \to k') : &= P(z_{im} = k' | z_{i(m-1)} = k, \alpha, r) \\
&: = \begin{cases} e^{-r_m d_m} + (1 - e^{-r_m d_m})\alpha_{k'm}, & k' = k \\ (1 - e^{-r_m d_m})\alpha_{k'm}, & k' \neq k \end{cases}
\end{aligned}
$$

where d_m is distance from $m - 1$ to m and $r_1, \ldots, r_M$ and the α_{km} parameters are to be estimated

The r parameters are related to local recombination rate and are the *per nucleotide* probability of a cluster switch

$P(h_i|\alpha, \theta, r)$ can be obtained by summing over all paths through the hidden states (local cluster membership) using the forward algorithm

Extending to genotype data

$$P_m(\{k_1, k_2\} \to \{k_1', k_2'\})$$

$$= \begin{cases} P_m(k_1 \to k_1')P_m(k_2 \to k_2') + P_m(k_1 \to k_2')P_m(k_2 \to k_1'), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k_1 \neq k_2 \text{ and } k_1' \neq k_2' \\ P_m(k_1 \to k_2')P_m(k_2 \to k_2'), \qquad\qquad \text{otherwise} \end{cases}$$

Extending to genotype data

$$P(g_{im}|z_{im} = \{k_1, k_2\}, \theta)$$

$$= \begin{cases} (1 - \theta_{k_1 m})(1 - \theta_{k_2 m}), & g_{im} = 0 \\ \theta_{k_1 m}(1 - \theta_{k_2 m}) + \theta_{k_2 m}(1 - \theta_{k_1 m}), & g_{im} = 1 \\ \theta_{k_2 m}\theta_{k_2 m}, & g_{im} = 2 \end{cases}$$