

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

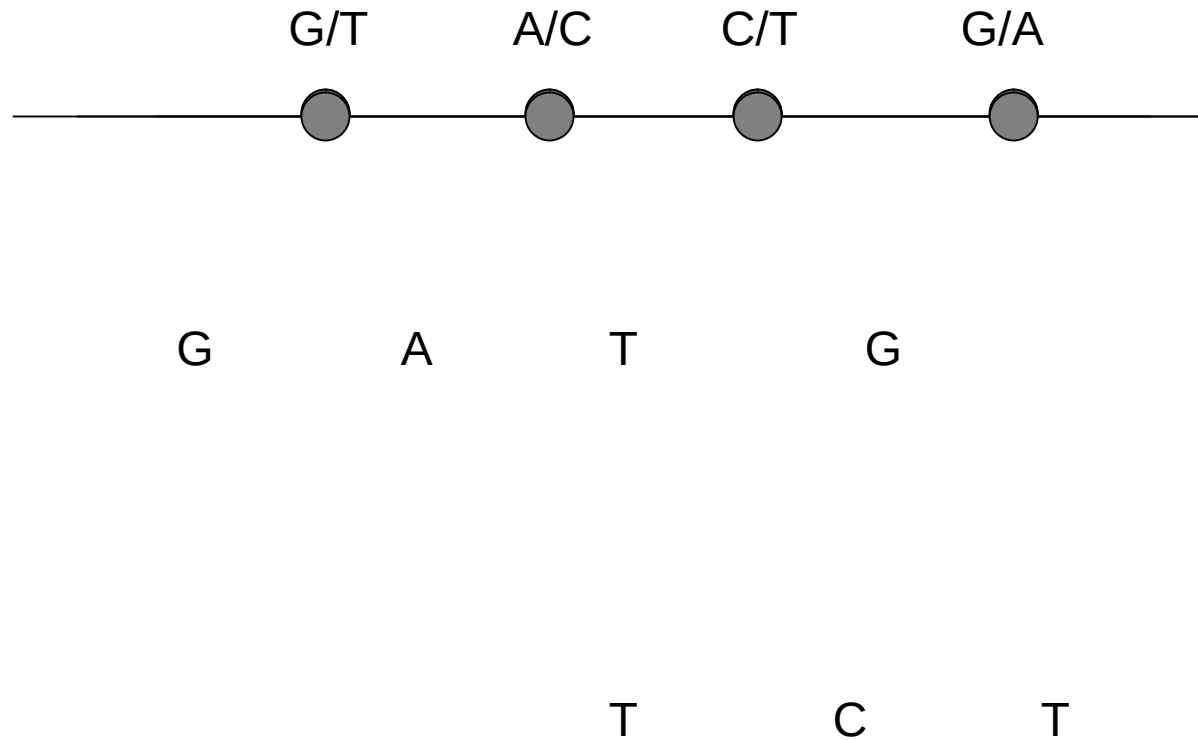
existence of haplotypes reduces the number of SNPs for genotyping

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)



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 $\rightarrow$ 0

YY $\rightarrow$ 1

XY $\rightarrow$ 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

<i>SNPs:</i>	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, $g_1 \dots g_n$ from n individuals. Consider the (unknown) set of haplotypes, $h_1 \dots h_n$ corresponding to these genotypes, occurring at frequencies $\theta_1 \dots \theta_n$.

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

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

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

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

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

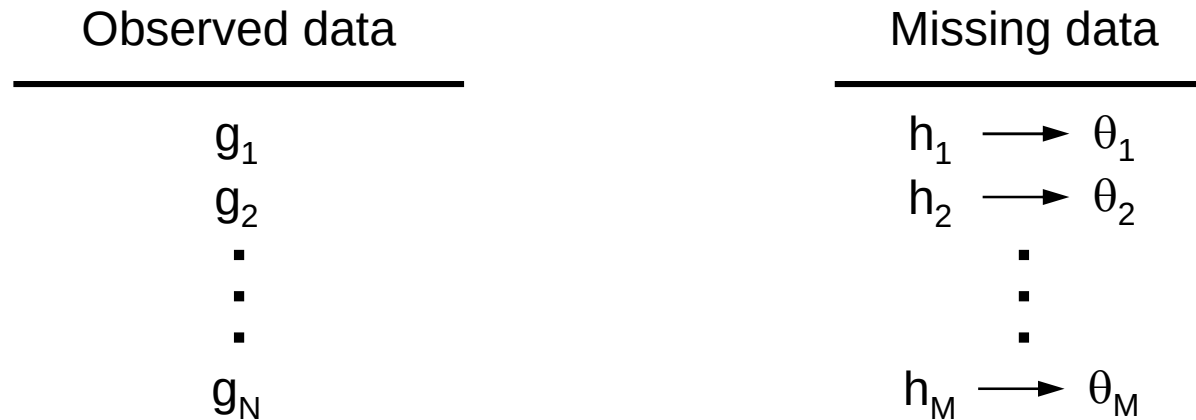
$$L(\theta) = P(\text{Data} | \theta) = P(g_1 | \theta)P(g_2 | \theta) \dots P(g_n | \theta)$$

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.

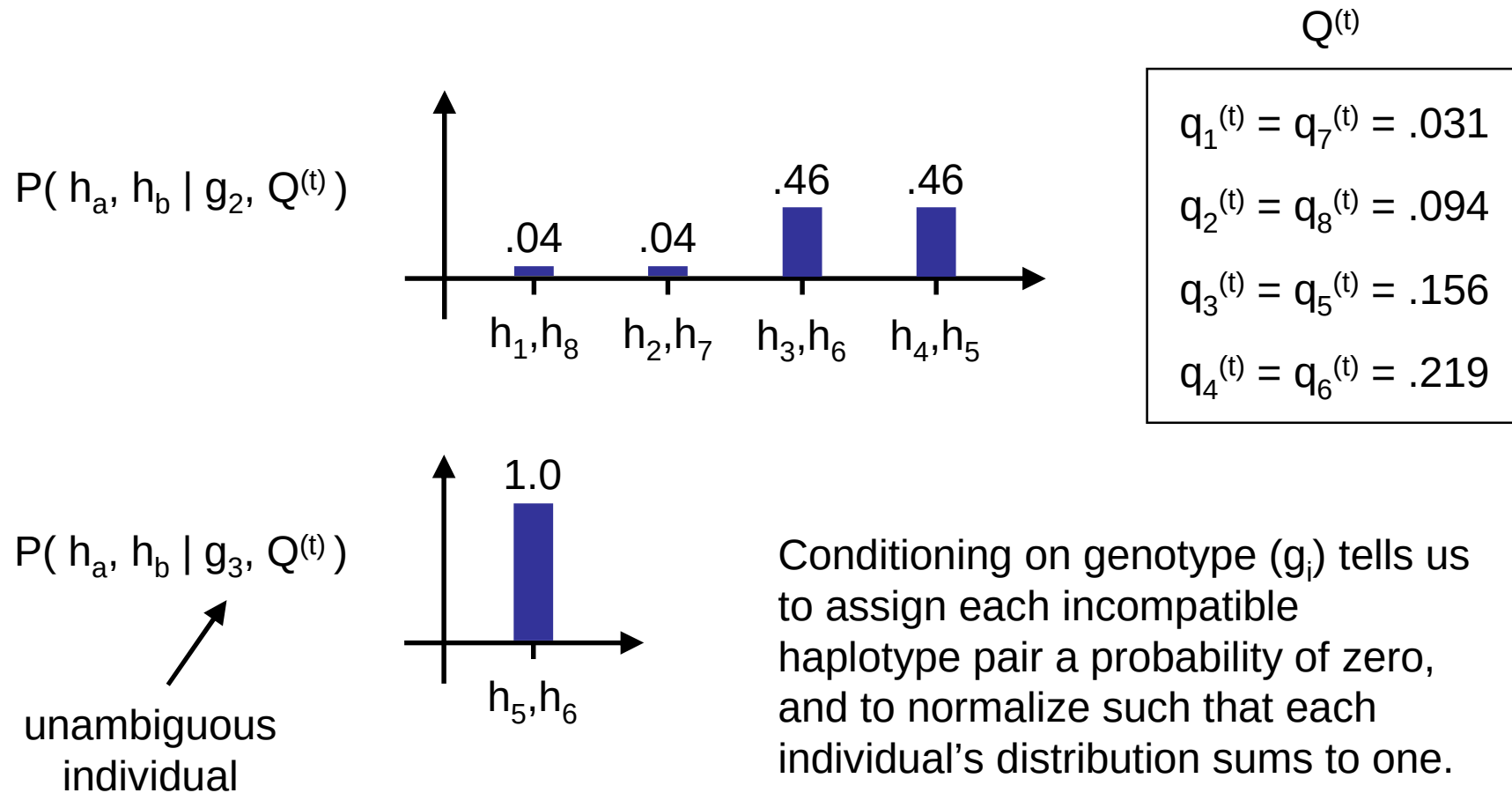


Initialization:

$$\begin{aligned}\theta_1 &= q_1 \\ \theta_2 &= q_2 \\ &\vdots \\ \theta_M &= q_M\end{aligned}\quad (\sum_M q_j = 1)$$

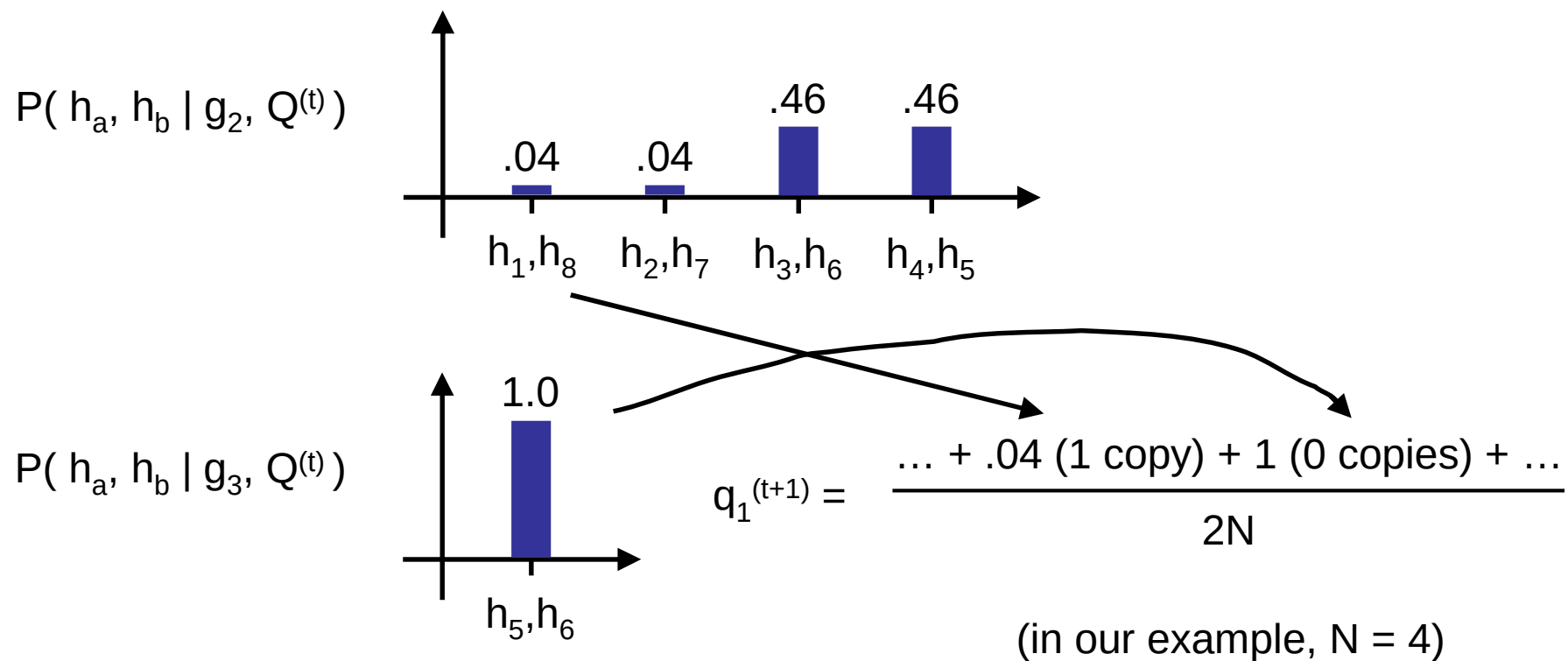
E(xpectation) step

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



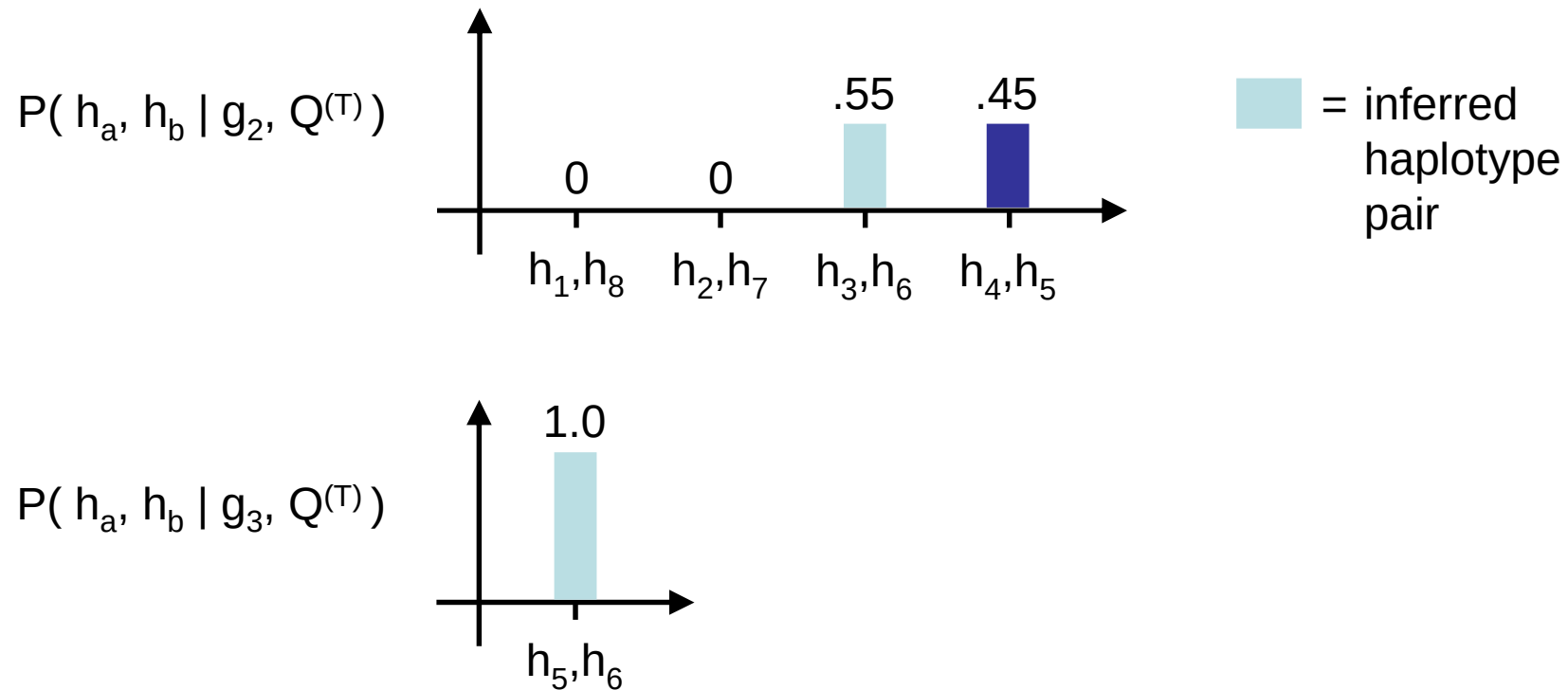
M(aximization) step

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 $P(h_a, h_b | g_i, Q^{(t)})$ 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:



Example: EM algorithm applied to the four genotypes seen earlier

[illegible]

Example: EM algorithm applied to the four genotypes seen earlier

	000	001	010	011	100	101	110	111	
Iteration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	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

	000	001	010	011	100	101	110	111	
Iteration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	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

	000	001	010	011	100	101	110	111	
Iteration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	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

	000	001	010	011	100	101	110	111	
Iteration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	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

	000	001	010	011	100	101	110	111	
Iteration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	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

	000	001	010	011	100	101	110	111	
Iteration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	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:

$$g_1 \quad \frac{0 \ 1 \ 1}{1 \ 0 \ 1}$$

$$g_2 \quad \frac{0 \ 1 \ 0}{1 \ 0 \ 1}$$

$$g_3 \quad \frac{1 \ 0 \ 0}{1 \ 0 \ 1}$$

$$g_4 \quad \frac{0 \ 1 \ 0}{0 \ 1 \ 1}$$

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

Method used by software package: PHASE

(see Stephens, Smith & Donnelly 2001)

Notation: let H represent a haplotype reconstruction for a set of n individuals; H_i is the haplotype reconstruction of individual i ; H_{-i} is the haplotype reconstruction of all individuals other than i .

Method used by software package: PHASE

(see Stephens, Smith & Donnelly 2001)

Construct a Markov chain that samples haplotype pairs

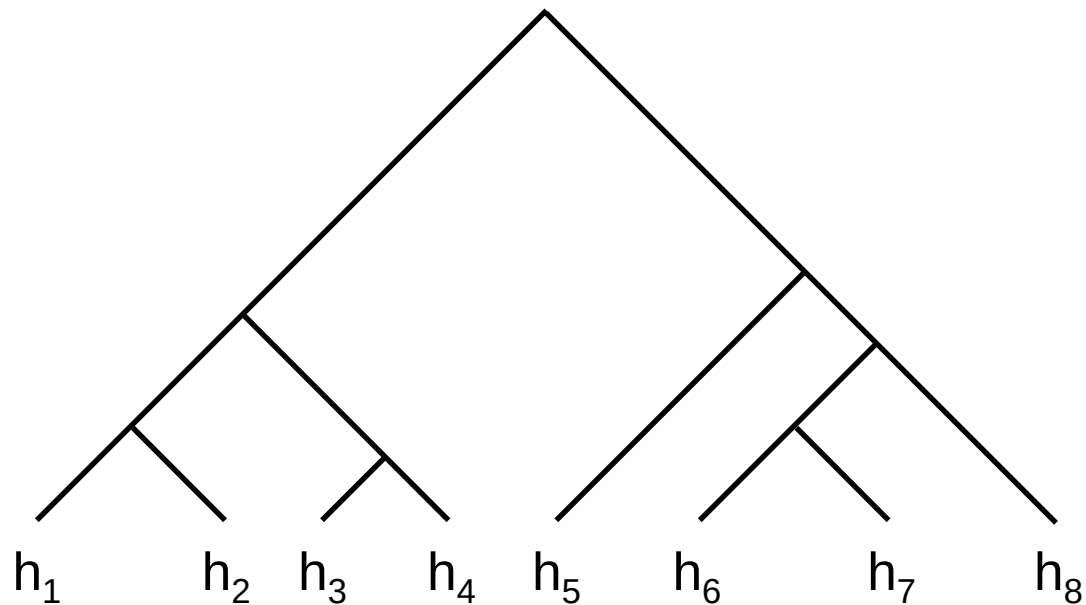
We would like to sample from the joint distribution of all haplotype pairs for all individuals

=> use Gibbs sampling

1. start with an arbitrary haplotype reconstruction $H^{(0)}$
2. select an individual, i , at random from all individuals with ambiguous haplotype pairs and sample $H_i^{(t+1)}$ from $P(H_i | G_i, H_{-i})$
3. set $H_j^{(t+1)} = H_j^{(t)}$ for all other individuals j

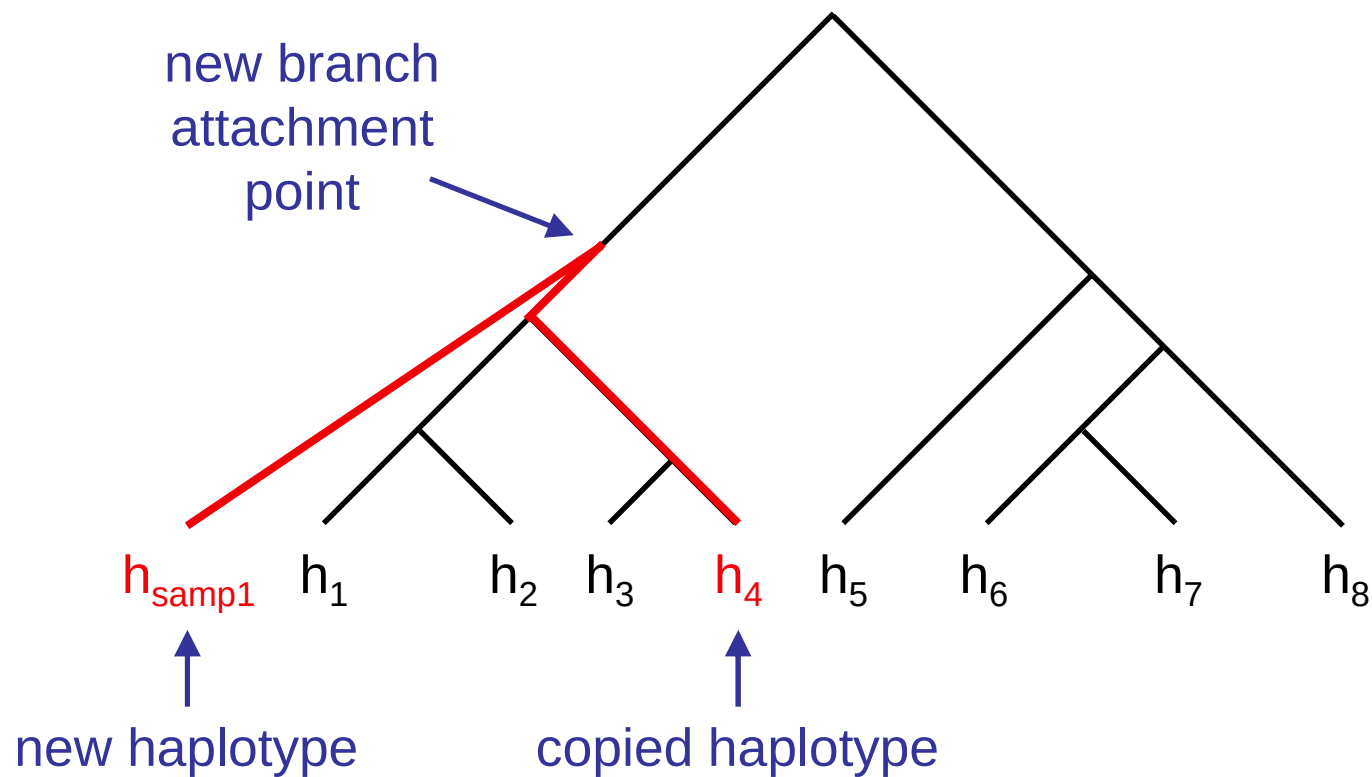
The PHASE algorithm: details

To see how Step 2 is accomplished, first suppose that the haplotypes of all individuals except individual i (i.e., $H_{-i}^{(t)}$) are related by the following tree:



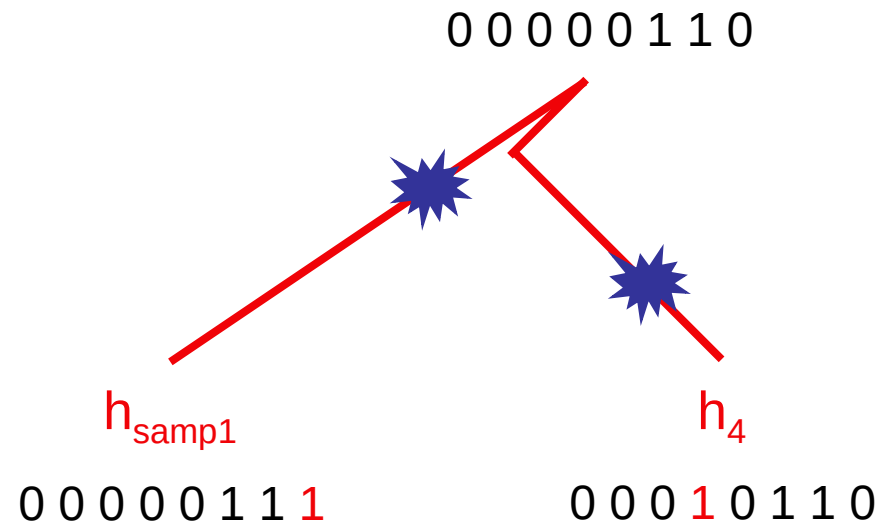
The PHASE algorithm: details

In sampling a new haplotype for individual i , PHASE chooses an existing haplotype at random (with probability equal to the proportion of that haplotype in H_{-i}) and copies it, possibly inexactly by inserting a new branch on the tree.



The PHASE algorithm: details

The vertical distance (time) between the haplotypes along the tree influences their similarity: the greater the distance, the greater the opportunity for chance mutations that could cause the sampled and copied haplotypes to differ. These distances tend to get shorter as the number of haplotypes in the tree increases.



Hidden Markov models have also been used to phase genotype data – e.g. fast PHASE (the following slides were not covered in lectures are not examinable)

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

Cluster model for haplotypes:

Given n haplotypes $(h_1, \dots, h_n)$ 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 practice 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

$$P(z_{i1} = k) = \alpha_{k1}$$

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

where d_m is distance from $m - 1$ to m and $r_1, \dots, r_M$ 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

$$P_m(\{k_1, k_2\} \rightarrow \{k'_1, k'_2\})$$
$$= \begin{cases} P_m(k_1 \rightarrow k'_1)P_m(k_2 \rightarrow k'_2) + P_m(k_1 \rightarrow k'_2)P_m(k_2 \rightarrow k'_1), & k_1 \neq k_2 \text{ and } k'_1 \neq k'_2 \\ P_m(k_1 \rightarrow k'_1)P_m(k_2 \rightarrow k'_2), & \text{otherwise} \end{cases}$$

Extending to genotype data

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