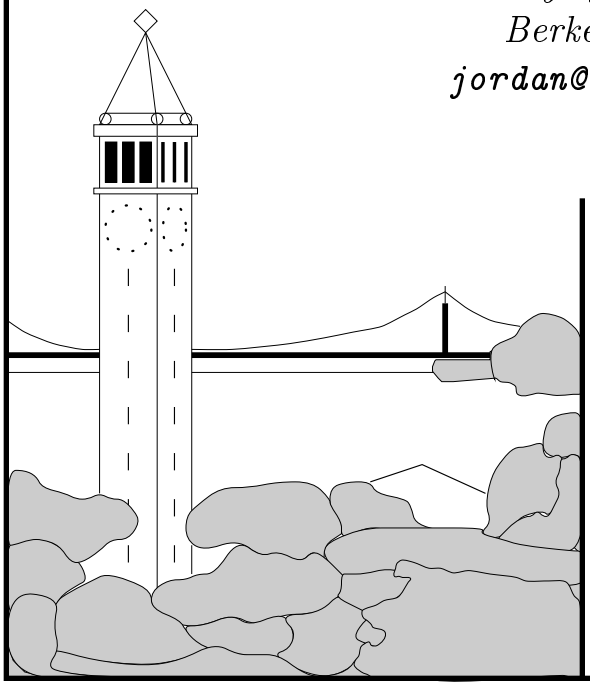
Bayesian Haplotype Inference via the Dirichlet Process

Eric P. Xing
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
epxing@cs.berkeley.edu

Roded Sharan
International Computer Science Institute
1947 Center St., Berkeley, CA 94704
roded@icsi.berkeley.edu

Michael I. Jordan
Computer Science and Statistics
University of California, Berkeley
Berkeley, CA 94720
jordan@cs.berkeley.edu



Report No. UCB/CSD-3-1275

September 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Bayesian Haplotype Inference via the Dirichlet Process

Eric P. Xing
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
epxing@cs.berkeley.edu

Roded Sharan
International Computer Science Institute
1947 Center St., Berkeley, CA 94704
roded@icsi.berkeley.edu

Michael I. Jordan
Computer Science and Statistics
University of California, Berkeley
Berkeley, CA 94720
jordan@cs.berkeley.edu

September 2003

Abstract

The problem of inferring haplotypes from genotypes of single nucleotide polymorphisms (SNPs) is essential for the understanding of genetic variation within and among populations, with important applications to the genetic analysis of disease propensities and other complex traits. In this paper we present a novel statistical model for haplotype inference. Our model is a Bayesian model based on a prior known as the Dirichlet process, a nonparametric prior which provides control over the size of the unknown pool of population haplotypes. The model also incorporates a likelihood that allows statistical errors in the haplotype/genotype relationship, trading off these errors against the size of the pool of haplotypes. We describe an algorithm based on Markov chain Monte Carlo for posterior inference. The overall result is a flexible Bayesian model that is reminiscent of parsimony methods in its preference for small haplotype pools. We apply this new approach to the analysis of both simulated and real genotype data, and compare to extant methods.

1 Introduction

The availability of a nearly complete human genome sequence makes it possible to begin to explore individual differences between DNA sequences on a genome-wide scale, and to search for associations of such genotypic variation with disease and other phenotypes [19]. Single nucleotide polymorphisms (SNPs) comprise the largest class of individual differences in DNA and have become a focus of research interest—millions of SNPs have thus far been detected out of an estimated total of ten million common SNPs [20, 23].

The list of alleles in contiguous sites in a local region of a single chromosome is called a *haplotype*. For diploid organisms, two haplotypes go together to make up a *genotype*, which is the list of unordered pairs of alleles in the region. That is, a genotype is obtained from a pair of haplotypes by omitting the specification of the association of each allele with one of the two chromosomes—its *phase*. Common typing methods yield the genotypes of a set of individuals, and typically do not provide phase information; the latter information can be obtained at a considerably higher cost [18]. It is therefore desirable to develop methods for inferring haplotypes from genotypes and possibly other data sources (e.g., pedigrees).

From the point of view of population genetics, the basic model underlying the haplotype inference problem is a finite mixture model. That is, letting $\mathcal{H}$ denote the set of all possible haplotypes associated with a given region (a set of cardinality 2^k in the case of binary polymorphisms, where k is the number of heterozygous sites), the probability of a genotype is given by:

$$p(g) = \sum_{h_1, h_2 \in \mathcal{H}} p(h_1, h_2) 1(h_1 \oplus h_2 = g), \tag{1}$$

where $1(h_1 \oplus h_2 = g)$ is the indicator function of the event that haplotypes h_1 and h_2 are consistent with g. Under the assumption of Hardy-Weinberg equilibrium (HWE), an assumption that is standard in the literature and will also be made here, the mixing proportion $p(h_1, h_2)$ is assumed to factor as $p(h_1)p(h_2)$.

Given this basic statistical structure, the simplest methodology for haplotype inference is maximum likelihood via the EM algorithm, treating the haplotype identities as latent variables and estimating the parameters $p(h)$ [7]. This methodology has rather severe computational requirements, in that a probability distribution must be maintained on the (large) set of possible haplotypes, but even more fundamentally it fails to capture the notion that small sets of haplotypes should be preferred. This notion derives from an underlying assumption that for relatively short regions there is limited diversity in a population due to population bottlenecks and relatively low rates of recombination and mutation.

One approach to dealing with this issue is to formulate a notion of "parsimony," and to develop algorithms that directly attempt to maximize parsimony. Several important papers have taken this approach [3, 12, 6, 22] and have yielded new insights and practical algorithms. Another approach is to elaborate the probabilistic model, in particular by incorporating priors on the parameters. Different priors have been discussed by different authors, ranging from simple Dirichlet priors [17] to priors based on the coalescent process [21] to priors that capture aspects of recombination [11]. These models provide implicit notions of parsimony, via the implicit "Ockham factor" of the Bayesian formalism [2].

Both parsimony-based and statistical approaches are useful in the case of phylogenetic inference [8], and we feel that it is likely that both will continue to play a role in haplotype inference as well. The approach that we take in the current paper is statistical, but we attempt to provide more explicit control over the number of inferred haplotypes than has been provided by the statistical methods proposed thus far, and the resulting inference algorithm has commonalities with the parsimony-based schemes.

Our approach is based on a nonparametric prior known as the *Dirichlet process* [9, 1]. In the setting of finite mixture models, the Dirichlet process—not to be confused with the Dirichlet distribution—is able to capture uncertainty about the number of mixture components [e.g., 5]. The basic setup can be explained in terms of an urn model, and a process that proceeds through data sequentially. Consider an urn which at the outset contains a ball of a single color. At each step we either draw a ball from the urn, and replace it with two balls of the same color, or we are given

a ball of a new color which we place in the urn, with a parameter defining the probabilities of these two possibilities. The association of data points to colors defines a "clustering" of the data. To make the link with Bayesian mixture models, we associate with each color a draw from the distribution defining the parameters of the mixture components.

This process defines a *prior distribution* for a mixture model with a random number of components. Multiplying this prior by a likelihood yields a *posterior distribution*. Markov chain Monte Carlo algorithms have been developed to sample from the posterior distributions associated with Dirichlet process priors [5, 16].

The usefulness of this framework for the haplotype problem should be clear—using a Dirichlet process prior we in essence maintain a pool of haplotype candidates that grows as observed genotypes are processed. The growth is controlled via a parameter in the prior distribution that corresponds to the choice of a new color in the urn model, and via the likelihood, which assesses the match of the new genotype to the available haplotypes.

To expand on this latter point, an advantage of the probabilistic formalism is its ability to elaborate the observation model for the genotypes to include the possibility of errors. In particular, the indicator function $1(h_1 \oplus h_2 = g)$ in Eq. (1) is suspect—there are many reasons why an individual genotype may not match with a current pool of haplotypes, such as the possibility of mutation or recombination in the meiosis for that individual, and errors in the genotyping or data recording process. Such sources of small differences should not lead to the inference procedure spawning new haplotypes.

In the current paper we present a statistical model for haplotype inference based on a Dirichlet process prior and a likelihood that includes error models for genotypes. We describe a Markov chain Monte Carlo procedure, in particular a procedure that makes use of both Gibbs and Metropolis-Hasting updates, for posterior inference. We present results of applying our method to the analysis of both simulated and real genotype data, comparing to the state-of-the-art PHASE algorithm [21]. On the simulated data our predictions are comparable to those obtained by PHASE. On a real dataset of [4] our results are again comparable to those of PHASE, and we outperform two other algorithms: HAP [13, 6] and HAPLOTYPER [17]. On data from [10], which is a difficult test case due to the small number of individuals in the sample, we outperform PHASE by a significant margin.

2 The Statistical Model

The input to a phasing algorithm can be represented as a *genotype matrix* G with columns corresponding to SNPs in their order along the chromosome and rows corresponding to genotyped individuals. $G_{j,i}$ represents the information on the two alleles of the i-th individual in SNP j. We denote the two alleles of a SNP by 0 and 1, and $G_{j,i}$ can take on one of four values: 0 or 1, indicating a homozygous site; 2, indicating a heterozygous site; and '?', indicating missing data. (Although we focus on binary data here, it is worth noting that our methods generalize immediately to non-binary data, and accommodate missing data).

We will describe our model in terms of a pool of ancestral haplotypes, or *templates*, from which each population haplotype originates [cf. 11]. The haplotype itself may undergo point mutation with respect to its template. The size of the pool and its composition are both unknown, and are treated as random variables under a Dirichlet process prior. We begin by providing a brief description of the Dirichlet process and subsequently show how this process can be incorporated into a model for haplotype inference.

2.1 The Dirichlet Process

Rather than present the Dirichlet process in full generality, we focus on the specific setting of a mixture model, and make use of the urn model to present the essential features of the process. For a fuller presentation, see, [e.g., 14]. We assume that data X arise from a mixture distribution with mixture components $p(x|\phi)$. We assume the existence of a *base measure* $G(\phi)$, which is one of the two parameters of the Dirichlet process. (The other is the parameter τ, which we present below). The parameter $G(\phi)$ is not the prior for ϕ, but is used to generate a prior for ϕ, in the manner that we now discuss.

Consider the following process for generating samples $\{x_1, x_2, \ldots, x_n\}$ from a mixture model consisting of an unspecified number of mixture components, or *equivalence classes*:

- The first sample x_1 is sampled from a distribution $p(x|\phi_1)$, where the parameter ϕ_1 is sampled from the base measure $G(\phi)$.
- The ith sample, x_i, is sampled from the distribution $p(x|\phi_{c_i})$, where:
  - The equivalence class of sample i, c_i, is drawn from the following distribution:

$$p(c_i = c_j \text{ for some } j < i | c_1, \ldots, c_{i-1}) = \frac{n_{c_j}}{i - 1 + \tau} \quad (2)$$

$$p(c_i \neq c_j \text{ for all } j < i | c_1, \ldots, c_{i-1}) = \frac{\tau}{i - 1 + \tau}, \quad (3)$$

    where n_{c_i} is the *occupancy number* of class c_i—the number of previous samples that belong to class c_i.
  - The parameter ϕ_{c_i} associated with the mixture component c_i is obtained as follows:

$$\phi_{c_i} = \phi_{c_j} \quad \text{if } c_i = c_j \text{ for some } j < i \text{ (i.e., } c_i \text{ is a previously populated equivalence class)}$$
$$\phi_{c_i} \sim G(\phi) \quad \text{if } c_i \neq c_j \text{ for all } j < i \text{ (i.e., } c_i \text{ is a new equivalence class).}$$

Eqs. (2) and (3) define a conditional prior for the equivalence class indicator c_i of each sample during a sequential sampling process. They imply a self-reinforcing property for the choice of equivalence class of each new sample—previously populated classes are more likely to be chosen.

It is important to emphasize that the process that we have discussed will be used as a *prior distribution*. We now embed this prior in a full model that includes a likelihood for the observed data. In Section 3 we develop Markov chain Monte Carlo (MCMC) inference procedures for this model.

2.2 The Model

We present a probabilistic model for the generation of haplotypes in a population and for the generation of genotypes from these haplotypes. We assume that each individual's genotype is formed by drawing two random *templates* from an ancestral pool, and that these templates are subject to random perturbation. The model is displayed as a graphical model (also known as a Bayesian network) in Figure 1.

Let J be an ordered list of loci of interest. For each individual i, we denote his/her paternal haplotype by $H_{i_0} := [H_{1,i_0}, \ldots, H_{J,i_0}]$ and maternal haplotype by $H_{i_1} := [H_{1,i_1}, \ldots, H_{J,i_1}]$. We denote a set of ancestral templates as $\mathbf{A} = \{A^{(1)}, A^{(2)}, \ldots\}$, where $A^{(k)} := [A_1^{(k)}, \ldots, A_J^{(k)}]$ is a particular member of this set. The set $\mathbf{A}$ is a random variable whose cardinality and composition

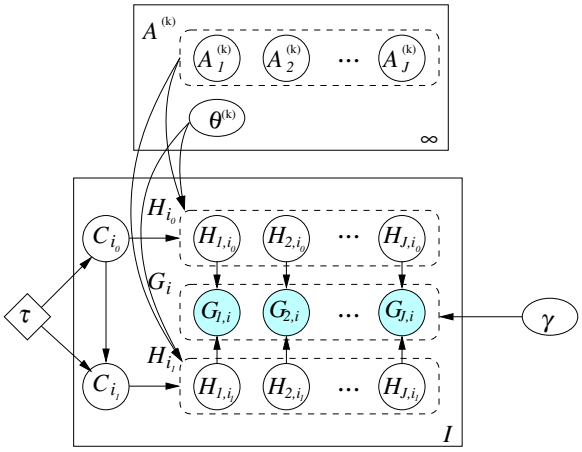


Figure 1: The graphical model representation of the haplotype model with a Dirichlet process prior. Circles represent the state variables, ovals represent the parameter variables, and diamonds represent fixed parameters. The dashed boxes denote sets of variables corresponding to the same ancestral template, haplotype, and genotype, respectively. The solid boxes corresponds to i.i.d. replicates of a set of variables, each associated with a particular individual (I copies), or ancestral template (an unbounded number of copies), respectively. Arrows between variables or boxes denote dependencies between variables or sets of variables.

are not fixed, but rather vary with realizations of the Dirichlet process and vary with the observed data.

In our framework, the probability distribution of the haplotype variable H_{i_t}, where the sub-subscript $t \in \{0, 1\}$ indexes paternal or maternal origin, is modeled by a mixture model with an unspecified number of mixture components, each corresponding to an equivalence class associated with a particular ancestor. For each individual i, we define the equivalence class variables C_{i_0} and C_{i_1} for the paternal and maternal haplotypes, respectively, to specify the ancestral origin of the corresponding haplotype. The C_{i_t} are the random variables corresponding to the equivalence classes of the Dirichlet process. The base measure G of the Dirichlet process is a joint measure on ancestral haplotypes A and mutation parameters θ, where the latter captures the probability that an allele at a locus is identical to the ancestor at this locus. We let $G(A, \theta) = p(A)p(\theta)$, and we assume that $p(A)$ is a uniform distribution over all possible haplotypes. We let $p(\theta)$ be a beta distribution $Beta(\alpha_h, \beta_h)$, and we choose a small value for $\beta_h/(\alpha_h + \beta_h)$, corresponding to a prior expectation of a low mutation rate.

Given C_{i_t} and a set of ancestors, we define the conditional probability of the corresponding haplotype instance $h := [h_1, \ldots, h_J]$ to be:

$$\begin{aligned} p(H_{i_t} = h | C_{i_t} = k, \mathbf{A} = \mathbf{a}, \boldsymbol{\theta}) &= p(H_{i_t} = h | A^{(k)} = a, \theta) \\ &= \prod_j p(h_j | a_j, \theta), \end{aligned} \tag{4}$$

where $p(h_j|a_j, \theta)$ is the probability of having allele h_j at locus j given its ancestor. Eq. (4) assumes that each locus is mutated independently with the same error rate. For haplotypes, H_{j,i_t} takes values from a set B of alleles. We use the following *single-locus mutation model*:

$$p(h_j|a_j, \theta) = \theta^{1(h_j = a_j)} \Big(\frac{1-\theta}{|B|-1} \Big)^{1(h_j \neq a_j)} \tag{5}$$

where $1(\cdot)$ is the indicator function.

The joint conditional distribution of haplotype instances $\mathbf{h} = \{h_{i_t} \,:\, t \in \{0,1\}, i \in \{1,2,\ldots,I\}\}$ and parameter instances $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$, given the ancestor indicator $\mathbf{c}$ of haplotype instances and the set of ancestors $\mathbf{a} = \{a^{(1)}, \ldots, a^{(K)}\}$, can be written explicitly as:

$$p(\mathbf{h}, \boldsymbol{\theta} | \mathbf{c}, \mathbf{a}) \;=\; \prod_k \theta_k^{m_k + \alpha_h - 1} \Big(\frac{1 - \theta_k}{|B| - 1} \Big)^{m'_k} \big[1 - \theta_k\big]^{\beta_h - 1} \tag{6}$$

where $m_k = \sum_j \sum_i \sum_t 1(h_{j,i_t} = a_j^{(k)}) 1(c_{i_t} = k)$ is the number of alleles that were not mutated with respect to the ancestral allele, and $m'_k = \sum_j \sum_i \sum_t 1(h_{j,i_t} \neq a_j^{(k)}) 1(c_{i_t} = k)$ is the number of mutated alleles. The count $\mathbf{m}_k = \{m_k, m'_k\}$ is a sufficient statistic for the parameter θ_k and the count $\mathbf{m} = \{\mathbf{m}_k, \mathbf{m}'_k\}$ is a sufficient statistic for the parameter $\boldsymbol{\theta}$. The marginal conditional distribution of haplotype instances can be obtained by integrating out θ in Eq. (6):

$$p(\mathbf{h} | \mathbf{c}, \mathbf{a}) \;=\; \prod_k R(\alpha_h, \beta_h) \frac{\Gamma(\alpha_h + m_k) \Gamma(\beta_h + m'_k)}{\Gamma(\alpha_h + \beta_h + m_k + m'_k)} \Big(\frac{1}{|B| - 1} \Big)^{m'_k}, \tag{7}$$

where $\Gamma(\cdot)$ is the *gamma* function, and $R(\alpha_h, \beta_h) = \frac{\Gamma(\alpha_h + \beta_h)}{\Gamma(\alpha_h)\Gamma(\beta_h)}$ is the normalization constant associated with $Beta(\alpha_h, \beta_h)$. (For simplicity, we use the abbreviation R_h for $R(\alpha_h, \beta_h)$ in the sequel).

We now introduce a *noisy observation model* for the genotypes. We let $G_i = [G_{1,i}, \ldots, G_{J,i}]$ denote the *joint genotype* of individual i at loci $[1, \ldots, J]$, where each $G_{j,i}$ denotes the *genotype* at locus j. We assume that the observed genotype at a locus is determined by the paternal and maternal alleles of this locus as follows:

$$p(g_{j,i} | h_{j,i_0}, h_{j,i_1}, \gamma) = \gamma^{1(h_{j,i} = g_{j,i})} [\mu_1 (1 - \gamma)]^{1(h_{j,i} \overset{1}{\neq} g_{j,i})} [\mu_2 (1 - \gamma)]^{1(h_{j,i} \overset{2}{\neq} g_{j,i})}, \tag{8}$$

where $h_{j,i} \triangleq h_{j,i_0} \oplus h_{j,i_1}$ denotes the unordered pair of two actual SNP allele instances at locus j; "$\overset{1}{\neq}$" denotes set difference by exactly one element (i.e., the observed genotype is heterozygous, while the true one is homozygous); "$\overset{2}{\neq}$" denotes set difference of both elements (i.e., the observed and true genotypes are different and both are homozygous); and μ_1 and μ_2 are appropriately defined normalizing constants. We place a beta prior $Beta(\alpha_g, \beta_g)$ on γ. Assuming independent and identical error models for each locus, the joint conditional probability of the entire genotype observation $\mathbf{g} = \{g_i \,:\, i \in \{1, 2, \ldots, I\}\}$ and parameter γ, given all haplotype instances is:

$$\begin{aligned} p(\mathbf{g}, \gamma | \mathbf{h}) \;&=\; \prod_i p(g_i, \gamma | h_{i_0}, h_{i_1}) \\ &=\; \gamma^{\alpha_g + u - 1} \big[1 - \gamma\big]^{\beta_g + u' + u'' - 1} \mu_1^{u'} \mu_2^{u''}, \end{aligned} \tag{9}$$

where the sufficient statistics $\mathbf{u} = \{u, u', u''\}$ are computed as $u = \sum_{i,j} 1(h_{j,i} = g_{j,i})$, $u' = \sum_{i,j} 1(h_{j,i} \overset{1}{\neq} g_{j,i})$, and $u'' = \sum_{i,j} 1(h_{j,i} \overset{2}{\neq} g_{j,i})$, respectively. Note that $u + u' + u'' = IJ$. To reflect an assumption that the observational error rate is low we set $\beta_g / (\alpha_g + \beta_g)$ to a small constant (0.001). Again, the marginal conditional distribution of $\mathbf{g}$ is computed by integrating out γ.

3 Markov chain Monte Carlo for Haplotype Inference

In this section, we describe a Gibbs sampling algorithm for exploring the posterior distribution under our model, including the latent ancestral pool. We also present a Metropolis-Hastings variant of this algorithm that appears to mix better in practice.

3.1 A Gibbs sampling algorithm

The Gibbs sampler draws samples of each random variable from a predictive distribution of the variable to be sampled given (previously sampled) values of all the remaining variables of the model. The variables needed in our algorithm are: c_{i_t}, the index of the ancestral template of a haplotype instance t of individual i; $a_j^{(k)}$, the allele pattern at the j-th locus of the k-th ancestral template; h_{j,i_t}, the t allele of the SNP at the j-th locus of individual i; and $g_{j,i}$, the genotype at locus j of individual i (the only observed variables in the model). All other variables in the model—θ and γ—are integrated out. The Gibbs sampler thus assesses the values of c_{i_t}, $a_j^{(k)}$ and h_{j,i_t}.

Conceptually, the Gibbs sampler alternates between two coupled stages. First, given the current values of the hidden haplotypes, we sample the c_{i_t} and subsequently $a_j^{(k)}$, which are associated with the Dirichlet process prior. Second, given the current state of the ancestral pool and the ancestral template assignment for each individual, we sample the h_{j,i_t} variables in the basic haplotype model.

In the first stage, the predictive distribution of c_{i_t} is:

$$\begin{aligned} p(c_{i_t} = k \mid \mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a}) &\propto p(c_{i_t} = k \mid \mathbf{c}_{[-i_t]}) p(h_{i_t} | a^{(k)}, \mathbf{c}, \mathbf{h}_{[-i_t],}) \\ &= \begin{cases} \frac{n_{[-i_t],k}}{n-1+\tau} p(h_{i_t} | a^{(k)}, \mathbf{m}_{[-i_t],k}) & \text{if } k = c_{i'_{t'}} \text{ for some } i'_{t'} \neq i_t \\ \frac{\tau}{n-1+\tau} \sum_{a'} p(h_{i_t} | a') p(a') & \text{if } k \neq c_{i'_{t'}} \text{ for all } i'_{t'} \neq i_t \end{cases}, \end{aligned} \tag{10}$$

where $[-i_t]$ denotes the set of indices excluding i_t; $n_{[-i_t],k}$ represents the number of $c_{i'_{t'}}$ for $i'_{t'} \neq i_t$ that are equal to k; n represents the total number of instances sampled so far; and $\mathbf{m}_{[-i_t],k}$ denote the m sufficient statistics associated with all haplotype instances originating from ancestor k, except h_{i_t}. This expression is simply Bayes theorem with $p(h_{i_t}|a^{(k)}, \mathbf{c}, \mathbf{h}_{[-i_t],})$ playing the role of the likelihood and $p(c_{i_t} = k \mid \mathbf{c}_{[-i_t]})$ playing the role of the prior. The likelihood $p(h_{i_t}|a^{(k)}, \mathbf{m}_{[-i_t],k})$ is obtained by integrating over the parameter θ_k, as in Eq. (7).

The conditional probability for a newly proposed equivalence class k that is not populated by any previous samples requires a summation over all possible ancestors: $p(h_{i_t}) = \sum_{a'} p(h_{i_t}|a')p(a')$. Since the gamma function does not factorize over loci, computing this summation takes time that is exponential in the number of loci. To skirt this problem we endow each locus with its own mutation parameter θ_{kj}, with all parameters admitting the same beta prior $Beta(\alpha_h, \beta_h)$. This gives rise to a closed-form formula for the summation and also for the normalization constant in Eq. (10). It is also arguably a more accurate reflection of reality.

Now we need to sample the ancestor template $a^{(k)}$, where k is the newly sampled ancestor index for c_{i_t}. When k is not equal to any other existing index $c_{i'_{t'}}$, a value for $a^{(k)}$ needs to be chosen from $p(A|h_{i_t})$, the posterior distribution of A based on the prior $p(A)$ and the single dependent haplotype h_{i_t}. On the other hand, if k is an equivalence class populated by previous samples of $c_{i'_{t'}}$, we draw a new value of $a^{(k)}$ from $p(A|h_{i_t}, \text{ s.t. } c_{i_t} = k)$. If after a new sample of c_{i_t}, a template is no longer associated with any haplotype instance, we remove this template from the pool. The predictive distribution for this Gibbs step is therefore:

$$p(a_j^{(k)} | h_{j,i_t} \text{ s.t. } c_{i_t} = k) \propto$$
$$\begin{cases} p(h_{j,i_t} | a_j^{(k)}) = \left(\frac{\alpha_h}{\alpha_h+\beta_h}\right)^{1(h_{j,i_t}=a_j^{(k)})} \left(\frac{\beta_h}{|B-1|(\alpha_h+\beta_h)}\right)^{1(h_{j,i_t}\neq a_j^{(k)})} & \text{if } k \text{ is not previously instantiated} \\ p(h_{j,i_t} \text{ s.t. } c_{i_t} = k | a_j^{(k)}) = \frac{\Gamma(\alpha_h+m_{j,k})\Gamma(\beta_h+m'_{j,k})}{\Gamma(\alpha_h+\beta_h+n_k)\cdot|B-1|^{m'_{j,k}}} & \text{if } k \text{ is previously instantiated} \end{cases}, \tag{11}$$

where $m_{j,k}$ (respectively, $m'_{j,k}$) is the number of allelic instances originated from ancestor k at locus

j that are identical to (respectively, different from) the ancestor, when the ancestor has the pattern $a_j^{(k)}$.

We now proceed to the second sampling stage, in which we sample the haplotypes h_{i_t}. We sample each h_{j,i_t}, for all j, i, t, sequentially according to the following predictive distribution:

$$\begin{aligned} p(h_{j,i_t}|\mathbf{h}_{[-(j,i)]}, h_{j,i_{\bar{t}}}, \mathbf{c}, \mathbf{a}, \mathbf{g}) &\propto p(g_i|h_{j,i_t}, h_{j,i_{\bar{t}}}, \mathbf{u}_{[-(j,i)]}) p(h_{j,i_t}|a_j^{(k)}, \mathbf{m}_{[-(j,i_t)],k}) \\ &= R_g \frac{\Gamma(\alpha_g + u)\Gamma(\beta_g + (u' + u''))}{\Gamma(\alpha_g + \beta_g + IJ)} [\mu_1]^{u'} [\mu_2]^{u''} \times R_h \frac{\Gamma(\alpha_h + m_{j,k})\Gamma(\beta_h + m'_{j,k})}{\Gamma(\alpha_h + \beta_h + n_k) \cdot |B-1|^{m'_{j,k}}}, \end{aligned} \quad (12)$$

where $[-(j, i_t)]$ denotes the set of indices excluding (j, i_t) and $m_{j,k} = m_{[-(j,i_t)],k} + 1(h_{j,i_t} = a_j^{(k)})$ (and similarly for the other sufficient statistics). Note that during each sampling step, we do not have to recompute the $\Gamma(\cdot)$, because the sufficient statistics are either not going to change (e.g., when the newly sampled h_{j,i_t} is the same as the old sample), or only going to change by one (e.g., when the newly sampled h_{j,i_1} results in a change of the allele). In such cases the new gamma function can be easily updated from the old one.

3.2 Metropolis-Hasting sampling algorithm

Note that for a long list of loci, a uniform $p(A)$ of all possible ancestral template patterns will render the probability of sampling a new ancestor infinitesimal, due to the small value of the smoothed marginal likelihood of any haplotype pattern h_{i_t}, as computed from Eq. (10). This could result in slow mixing.

An alternative sampling strategy is to use a partial Gibbs sampling strategy with the following Metropolis-Hasting updates. For the proposal distribution for the equivalence class of h_{i_t} we use:

$$q(c^*_{i_t} = k|c_{i_t}) = \begin{cases} \frac{n_{[-i_t],k}}{n-1+\tau} & : \quad \text{if } k = c_{i'_{t'}} \text{ for some } i'_{t'} \neq i^t \\ \frac{\tau}{n-1+\tau} & : \quad \text{if } k \neq c_{i'_{t'}} \text{ for all } i'_{t'} \neq i_t \end{cases}. \quad (13)$$

Then we sample $a^{(c^*_{i_t})}$ sequentially according to Eq. (11). For target distribution $p(c_{i_t} = k|\mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a})$, the proposal factor cancels when computing the acceptance probability ξ, leaving:

$$\xi(c^*_{i_t}, c_{i_t}) = \min\Big[1, \frac{p(h_{i_t}|a^{(c^*_{i_t})})}{p(h_{i_t}|a^{(c_{i_t})})}\Big]. \quad (14)$$

In practice, we found that the above modification to the Gibbs sampling algorithm leads to substantial improvement of efficiency for long haplotype lists, whereas for short lists, the Gibbs sampler remains better due to the high (100%) acceptance rate.

4 Experimental Results

We validated our algorithm by applying it to simulated and real data and compared its performance to that of the state-of-the-art PHASE algorithm [21] and other current algorithms. We report on the results of both variants of our algorithm: The Gibbs sampler, denoted DP(Gibbs), and the Metropolis-Hasting sampler, denoted DP(MH). Throughout the experiments, we set the hyperparameter τ in the Dirichlet process to be roughly 1% of the population size, i.e., for a data set of 100 individuals, $\tau = 1$. We used a burn-in of 2000 iterations (or 4000 for datasets with more than 50 individuals), and used the next 6000 iterations for estimation.

4.1 Simulated data

In our first set of experiments we applied our method to simulated data from [21, "short sequence data"]. This data contains sets of $2n$ haplotypes, randomly paired to form n genotypes, under an infinite-sites model with parameters $\eta = 4$ and $R = 4$ determining the mutation and recombination rates, respectively (see [21] for additional details). We used the first 40 datasets for each combination of individuals and sites, where the number of individuals ranged between 10 and 50, and the number of sites ranged between 5 and 30.

To evaluate the performance of the algorithms we used the following error measures: err_s, the ratio of incorrectly phased SNP sites over all non-trivial heterozygous SNPs (excluding individuals with a single heterozygous SNP); err_i, the the ratio of incorrectly phased individuals over all non-trivial heterogeneous individuals; and d_s, the *switch distance*, which is the number of phase flips required to correct the predicted haplotypes over all non-trivial heterogeneous SNPs. The results are summarized in Table 1. Overall, we perform slightly worse than PHASE on the first two measures, and slightly better on the switch distance measure.

	DP(MH)			PHASE		
#individuals	err_s	err_i	d_s	err_s	err_i	d_s
10	0.060	0.216	0.051	0.046	0.182	0.054
20	0.039	0.152	0.039	0.029	0.136	0.046
30	0.036	0.121	0.038	0.024	0.101	0.027
40	0.030	0.094	0.029	0.019	0.071	0.026
50	0.028	0.082	0.024	0.019	0.072	0.025

Table 1: Performance results on simulated data from [21].

4.2 Real data

We also applied our algorithm to two real datasets and compared its performance to that of PHASE [21] and other algorithms.

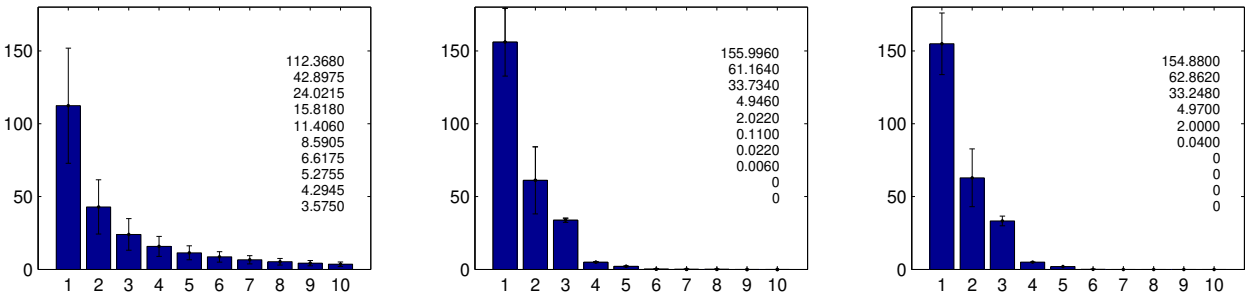
The first dataset contains the genotypes of 129 individuals over 103 polymorphic sites [4]. In addition it contains the genotypes of the parents of each individual, which allows the inference of a large portion of the haplotypes as in [6]. The performance results are summarized in Table 2. From Table 2, it is apparent that the Metropolis-Hasting sampling algorithm significantly outperforms the Gibbs sampler, and is to be preferred given the relatively limited number of sampling steps (~ 6000). The overall performance is comparable to that of PHASE and better than both HAP [13, 6] and HAPLOTYPER [17].

It is important to emphasize that our methods also provide a posteriori estimates of the ancestral pool of haplotype templates and their frequencies. We omit a listing of these haplotypes, but provide an illustrative summary of the evolution of these estimates during sampling (Figure 2).

The second dataset contains genotype data from four populations, 90 individuals each, across several genomic regions [10]. We focused the Yoruban population (D), which contains 30 trios of genotypes (allowing us to infer most of the true haplotypes) and analyzed the genotypes of 28 individuals over four medium-sized regions (see below). The results are summarized in Table 3. All methods yield higher error rates on these data, compared to the analysis of the data of [4], presumably due to the low sample size. In this setting, over all but one of the four regions, our algorithm outperformed PHASE for all three types of error measures. A preliminary analysis suggests that our performance gain may be due to the bias toward parsimony induced by the Dirichlet process prior. We found that the number of template haplotypes in our algorithm is typically small, whereas in PHASE, the haplotype pool can be very large (i.e., region 7b has 83 haplotypes, compared to 10 templates in our case and 28 individuals overall).

		DP(Gibbs)			DP(MH)			PHASE			HAP	HAPLOTYPER
block id.	length	err_s	err_i	d_s	err_s	err_i	d_s	err_s	err_i	d_s	err_s	err_s
1	14	0.223	0.485	0.229	0	0	0	0.003	0.030	0.003	0.007	0.039
2	5	0	0	0	0.007	0.026	0.007	0.007	0.026	0.007	0.036	0.065
3	5	0	0	0	0	0	0	0	0	0	0	0.008
4	11	0.143	0.262	0.128	0	0	0	0	0	0	0.015	-
5	9	0.020	0.066	0.020	0.011	0.033	0.011	0.011	0.033	0.011	0.027	0.151
6	27	0.071	0.191	0.074	0.005	0.043	0.005	0	0	0	0.018	0.041
7	7	0.005	0.018	0.005	0.005	0.018	0.005	0.005	0.018	0.005	0.068	0.214
8	4	0	0	0	0	0	0	0	0	0	0	0.252
9	5	0.029	0.097	0.029	0.012	0.032	0.012	0.012	0.032	0.012	0.057	0.152
10	4	0.007	0.025	0.007	0.007	0.025	0.007	0.008	0.025	0.008	0.042	0.056
11	7	0.010	0.034	0.005	0.005	0.017	0.005	0.011	0.034	0.011	0.033	0.093
12	5	0.010	0.037	0.020	0	0	0	0	0	0	0	0.077

Table 2: Performance results on the data of Daly et al. [4], using the block structure provided by [13]. The results of HAP and HAPLOTYPER are adapted from [13]. Since the error rate in [13] uses the number of both heterozygous and missing genotypes as the denominator, whereas we used only the non-trivial heterozygous ones, we rescaled the error rates of the two latter methods to be comparable to ours.



(a) (b) (c)

Figure 2: The top ten ancestral templates during Metropolis-Hasting sampling for block 1 of the data of Daly et al. [4]. (The numbers in the panels are the posterior means of the frequency of each template). (a) Immediately after burn-in (first 2000 samples). (b) 3000 samples after burn-in. (c) 6000 samples after burn-in.

		DP(MH)			PHASE		
region	length	err_s	err_i	d_s	err_s	err_i	d_s
16a	13	0.185	0.480	0.141	0.174	0.440	0.130
1b	16	0.100	0.250	0.160	0.200	0.450	0.180
25a	14	0.135	0.353	0.115	0.212	0.588	0.212
7b	13	0.105	0.278	0.066	0.145	0.444	0.092

Table 3: Performance on the data of [10].

5 Conclusion

We have proposed a Bayesian approach to the modeling of genotypes based on a Dirichlet process prior. We have shown that the Dirichlet process provides a natural representation of uncertainty regarding the size and composition of the pool of haplotypes underlying a population. Using Markov chain Monte Carlo algorithms, we have shown that this model leads to effective inference procedures for inference of the ancestral pool and for haplotype phasing based on a set of genotypes. The model accommodates growing data collections and noisy and/or incomplete observations. The approach also naturally imposes an implicit bias toward small ancestral pools during inference, reminiscent of parsimony methods, doing so in a well-founded statistical framework that permits errors.

Our focus here has been on adapting the technology of the Dirichlet process in the setting of the standard haplotype phasing problem. But an important underlying motivation for our work, and a general motivation for pursuing probabilistic approaches to genomic inference problems, is the potential value of our model as a building block for more expressive models. In particular, as in [11] and [15], the graphical model formalism naturally accommodates various extensions, such as segmentation of chromosomes into haplotype blocks and the inclusion of pedigree relationships. The Dirichlet process parameterization also provides a natural upgrade path for the considering of richer models; in particular, it is possible to incorporate more elaborate base measures G into the Dirichlet process framework—the coalescence-based distribution of [21] would be an interesting choice.

Acknowledgments

This research was supported in part by NSF ITR Grant CCR-0121555.

References

[1] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1973.

[2] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory.* Wiley, 1994.

[3] A. Clark et al. Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase. *Am J. Human Genetics*, 63:595–612, 1998.

[4] M.J. Daly et al. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232, 2001.

[5] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *J. Am. Statist. Assoc.*, 90:577–588, 2002.

[6] E. Eskin, E. Halperin, and R.M. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of Bioinformatics and Computational Biology*, 1:1–20, 2003.

[7] L Excoffier and M Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–7, 1995.

[8] J. Felsenstein. *Inferring Phylogenies.* Sinauer Associates, Inc., 2003.

[9] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230, 1973.

[10] S. B. Gabriel et al. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.

[11] D. Greenspan and D. Geiger. Model-based inference of haplotype block variation. In *Proceedings of the 7th International Conference on Computational Molecular Biology (RECOMB 2003)*, 2003.

[12] D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions (extended abstract). In *Proceedings of the 6th International Conference on Computational Molecular Biology (RECOMB 2002)*, pages 166–175, 2002.

[13] E. Halperin and E. Eskin. Haplotype reconstruction from genotype data using imperfect phylogeny. TR, CS Dept. Columbia University, 2002.

[14] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *J. Am. Statist. Assoc.*, 90:161–173, 2001.

[15] S. L. Lauritzen and N. A. Sheehan. Graphical models for genetic analysis. TR R-02-2020, Aalborg University, 2002.

[16] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *J. Computational and Graphical Statistics*, 9(2):249–256, 2000.

[17] T. Niu, S. Qin, X. Xu, and J. Liu. Bayesian haplotype inference for multiple linked single nucleotide polymorphisms. *Am. J. Hum.Genet.*, 70:157–169, 2002.

[18] N. Patil et al. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

[19] N. J. Risch. Searching for genetic determinants in the new millennium. *Nature*, 405(6788):847–56, 2000.

[20] R. Sachidanandam et al. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 291:1298–2302, 2001.

[21] M. Stephens, N. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989, 2001.

[22] Bafna V, Halldorsson BV, Schwartz R, Clark AG, and Istrail S. Haplotypes and informative snp selection algorithms: Don't block out information. In *Proceedings of the 7th International Conference on Computational Molecular Biology (RECOMB 2003)*, pages 19–27, 2002.

[23] C. Venter et al. The sequence of the human genome. *Science*, 291:1304–51, 2001.